Información general de la instalación

Astra Control Center

NetApp
March 12, 2024

Tabla de contenidos

Información general de la instalación 1
Instale Astra Control Center mediante el proceso estándar 1
Instale Astra Control Center utilizando OpenShift OperatorHub 43
Instale Astra Control Center con un back-end de almacenamiento de Cloud Volumes ONTAP 53
Configurar Astra Control Center después de la instalación 69

Información general de la instalación

Elija y complete uno de los siguientes procedimientos de instalación de Astra Control Center:

- "Instale Astra Control Center mediante el proceso estándar"
- "(Si utiliza Red Hat OpenShift) instale Astra Control Center mediante OpenShift OperatorHub"
- "Instale Astra Control Center con un back-end de almacenamiento de Cloud Volumes ONTAP"

En función de su entorno, es posible que se necesite una configuración adicional después de instalar Astra Control Center:

- "Configurar Astra Control Center después de la instalación"

Instale Astra Control Center mediante el proceso estándar

Para instalar Astra Control Center, descargue el paquete de instalación desde el sitio de soporte de NetApp y realice los siguientes pasos. Puede utilizar este procedimiento para instalar Astra Control Center en entornos conectados a Internet o con conexión por aire.

Expanda para otros procedimientos de instalación

- **Instalar con Red Hat OpenShift OperatorHub**: Utilice esto "procedimiento alternativo" Para instalar Astra Control Center en OpenShift mediante OperatorHub.
- **Instalar en la nube pública con Cloud Volumes ONTAP backend**: Uso "estos procedimientos" Para instalar Astra Control Center en Amazon Web Services (AWS), Google Cloud Platform (GCP) o Microsoft Azure con un back-end de almacenamiento de Cloud Volumes ONTAP.

Para ver una demostración del proceso de instalación de Astra Control Center, consulte "este vídeo".

Antes de empezar

- **Cumplir con los requisitos ambientales**: "Antes de comenzar la instalación, prepare su entorno para la implementación de Astra Control Center".

  Pon en marcha Astra Control Center en un tercer dominio de fallo o sitio secundario. Esto se recomienda para la replicación de aplicaciones y la recuperación ante desastres fluida.

- **Asegurar servicios saludables**: Comprueba que todos los servicios API estén en buen estado y disponibles:

  ```
  kubectl get apiservices
  ```

- **Asegúrese de que un FQDN enrutable**: El FQDN de Astra que planea utilizar se puede enrutar al clúster. Esto significa que tiene una entrada DNS en el servidor DNS interno o que está utilizando una ruta URL principal que ya está registrada.
- **Configurar gestor de cert**: Si ya existe un gestor de cert en el clúster, debe realizar algunos "requisitos previos" Por lo tanto, Astra Control Center no intenta instalar su propio administrador de certificados. De

forma predeterminada, Astra Control Center instala su propio administrador de certificados durante la instalación.

- **Acceda al registro de imágenes de NetApp Astra Control**:
  Tiene la opción de obtener imágenes de instalación y mejoras de funcionalidades para Astra Control, como Astra Control Provisioner, desde el registro de imágenes de NetApp.

  Expanda para obtener los pasos

  a. Registra tu ID de cuenta de Astra Control que tendrás que iniciar sesión en el registro.

     Puedes ver tu ID de cuenta en la interfaz de usuario web de Astra Control Service. Selecciona el icono de la figura en la parte superior derecha de la página, selecciona **Acceso API** y escribe tu ID de cuenta.

  b. En la misma página, selecciona **Generar token de API** y copia la cadena de token de API en el portapapeles y guárdalo en tu editor.

  c. Inicia sesión en el Registro de Astra Control:

     ```
     docker login cr.astra.netapp.io -u <account-id> -p <api-token>
     ```

- **Considera una malla de servicio**: Se recomienda encarecidamente que los canales de comunicaciones del clúster host de Astra Control estén protegidos mediante un "malla de servicio compatible".

Detalles de malla de servicio de Istio

Para el uso de la malla de servicio de Istio, deberá hacer lo siguiente:

- Agregue un `istio-injection:enabled` etiqueta En el espacio de nombres de Astra antes de poner en marcha Astra Control Center.
- Utilice la `Generic` ajuste de entrada y proporcionar una entrada alternativa para equilibrio de carga externo.
- Para los clústeres de Red Hat OpenShift, debe definirlos `NetworkAttachmentDefinition` En todos los espacios de nombres del Centro de control de Astra asociados (`netapp-acc-operator`, `netapp-acc`, `netapp-monitoring` para clusters de aplicaciones o cualquier espacio de nombres personalizado que se haya sustituido).

```
cat <<EOF | oc -n netapp-acc-operator create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-acc create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-monitoring create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF
```

- **Solo controlador SAN de ONTAP**: Si está utilizando un controlador SAN de ONTAP, asegúrese de que la función multivía esté habilitada en todos sus clústeres de Kubernetes.

Pasos

Para instalar Astra Control Center, lleve a cabo los siguientes pasos:

- Descargue y extraiga Astra Control Center
- Instale el complemento Astra kubectl de NetApp
- Agregue las imágenes al registro local
- Configurar espacio de nombres y secreto para registros con requisitos de autenticación

- Instale el operador de Astra Control Center
- Configurar Astra Control Center
- Complete la instalación del centro de control de Astra y del operador
- Comprobar el estado del sistema
- Configure la entrada para el equilibrio de carga
- Inicie sesión en la interfaz de usuario de Astra Control Center

> No elimine el operador Astra Control Center (por ejemplo, `kubectl delete -f astra_control_center_operator_deploy.yaml`) En cualquier momento durante la instalación o el funcionamiento de Astra Control Center para evitar la eliminación de las dosis.

Descargue y extraiga Astra Control Center

Puede elegir descargar el paquete Astra Control Center desde el sitio de soporte de NetApp o utilizar Docker para extraer el paquete del registro de imágenes del servicio de control de Astra.

Sitio de soporte de NetApp

1. Descargue el paquete que contiene Astra Control Center (`astra-control-center-[version].tar.gz`) del "Página de descargas de Astra Control Center".
2. (Recomendado pero opcional) Descargue el paquete de certificados y firmas para Astra Control Center (`astra-control-center-certs-[version].tar.gz`) para verificar la firma del paquete.

   Amplíe para obtener más detalles

   ```
   tar -vxzf astra-control-center-certs-[version].tar.gz
   ```

   ```
   openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub
   -signature certs/astra-control-center-[version].tar.gz.sig
   astra-control-center-[version].tar.gz
   ```

   Se mostrará la salida `Verified OK` después de una verificación correcta.

3. Extraiga las imágenes del paquete Astra Control Center:

   ```
   tar -vxzf astra-control-center-[version].tar.gz
   ```

Registro de imágenes de Astra Control

1. Inicia sesión en el servicio Astra Control.
2. En el Dashboard, selecciona **Desplegar una instancia autogestionada de Astra Control**.
3. Sigue las instrucciones para iniciar sesión en el registro de imágenes de Astra Control, extraer la imagen de instalación de Astra Control Center y extraer la imagen.

Instale el complemento Astra kubectl de NetApp

Puede utilizar el complemento de línea de comandos kubectl de Astra de NetApp para insertar imágenes en un repositorio de Docker local.

Antes de empezar

NetApp proporciona binarios de complementos para diferentes arquitecturas de CPU y sistemas operativos. Debe saber qué CPU y sistema operativo tiene antes de realizar esta tarea.

Si ya tiene instalado el plugin desde una instalación anterior, "asegúrese de tener la versión más reciente" antes de realizar estos pasos.

Pasos

1. Enumera los binarios para complementos de kubectl de Astra de NetApp disponibles:

La biblioteca de complementos kubectl forma parte del paquete tar y se extrae en la carpeta `kubectl-astra`.

```
ls kubectl-astra/
```

2. Mueva el archivo que necesita para su sistema operativo y la arquitectura de CPU a la ruta actual y cámbiele el nombre a. `kubectl-astra`:

```
cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
```

Agregue las imágenes al registro local

1. Complete la secuencia de pasos apropiada para el motor del contenedor:

Docker

1. Cambie al directorio raíz del tarball. Debería ver el `acc.manifest.bundle.yaml` archivo y estos directorios:

   ```
   acc/
   kubectl-astra/
   acc.manifest.bundle.yaml
   ```

2. Inserte las imágenes del paquete en el directorio de imágenes de Astra Control Center en su registro local. Realice las siguientes sustituciones antes de ejecutar el `push-images` comando:

   - Sustituya <BUNDLE_FILE> por el nombre del archivo Astra Control Bundle (`acc.manifest.bundle.yaml`).
   - Sustituya <MY_FULL_REGISTRY_PATH> por la URL del repositorio de Docker; por ejemplo, "<a href="https://<docker-registry>"" class="bare">https://<docker-registry>".
   - Reemplace <MY_REGISTRY_USER> por el nombre de usuario.
   - Sustituya <MY_REGISTRY_TOKEN> por un token autorizado para el registro.

     ```
     kubectl astra packages push-images -m <BUNDLE_FILE> -r
     <MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
     <MY_REGISTRY_TOKEN>
     ```

Podman

1. Cambie al directorio raíz del tarball. Debería ver este archivo y directorio:

   ```
   acc/
   kubectl-astra/
   acc.manifest.bundle.yaml
   ```

2. Inicie sesión en su registro:

   ```
   podman login <YOUR_REGISTRY>
   ```

3. Prepare y ejecute una de las siguientes secuencias de comandos personalizadas para la versión de Podman que utilice. Sustituya <MY_FULL_REGISTRY_PATH> por la URL del repositorio que incluye cualquier subdirectorio.

   ```
   <strong>Podman 4</strong>
   ```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.10.0-68
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/${
PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.10.0-68
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/${
PACKAGEVERSION}/${astraImageNoPath}
done
```

La ruta de acceso de imagen que crea el script debe parecerse a la siguiente, dependiendo de la configuración del Registro:

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/23.10.0-68/image:version
```

Configurar espacio de nombres y secreto para registros con requisitos de autenticación

1. Exporte el comando kubeconfig del clúster de hosts de Astra Control Center:

```
export KUBECONFIG=[file path]
```

Antes de completar la instalación, asegúrese de que su kubeconfig apunte al clúster donde desea instalar Astra Control Center.

2. Si utiliza un registro que requiere autenticación, debe hacer lo siguiente:

Expanda para obtener los pasos

a. Cree el `netapp-acc-operator` espacio de nombres:

```
kubectl create ns netapp-acc-operator
```

b. Cree un secreto para `netapp-acc-operator` espacio de nombres. Añada información sobre Docker y ejecute el siguiente comando:

El marcador de posición `your_registry_path` debe coincidir con la ubicación de las imágenes que ha cargado anteriormente (por ejemplo, `[Registry_URL]/netapp/astra/astracc/23.10.0-68`).

```
kubectl create secret docker-registry astra-registry-cred -n
netapp-acc-operator --docker-server=[your_registry_path] --docker
-username=[username] --docker-password=[token]
```

Si elimina el espacio de nombres después de que se genere el secreto, vuelva a crear el espacio de nombres y, a continuación, vuelva a generar el secreto para el espacio de nombres.

c. Cree el `netapp-acc` (o nombre personalizado).

```
kubectl create ns [netapp-acc or custom namespace]
```

d. Cree un secreto para `netapp-acc` (o nombre personalizado). Añada información sobre Docker y ejecute el siguiente comando:

```
kubectl create secret docker-registry astra-registry-cred -n
[netapp-acc or custom namespace] --docker
-server=[your_registry_path] --docker-username=[username]
--docker-password=[token]
```

Instale el operador de Astra Control Center

1. Cambie el directorio:

```
cd manifests
```

2. Edite la implementación del operador de Astra Control Center YAML (`astra_control_center_operator_deploy.yaml`) para referirse a su registro local y secreto.

```
vim astra_control_center_operator_deploy.yaml
```

> Una muestra anotada de AYLMA sigue estos pasos.

a. Si utiliza un registro que requiere autenticación, reemplace la línea predeterminada de `imagePullSecrets: []` con lo siguiente:

```
imagePullSecrets: [{name: astra-registry-cred}]
```

b. Cambiar `ASTRA_IMAGE_REGISTRY` para la `kube-rbac-proxy` imagen a la ruta del registro en la que se insertó la imagen en un paso anterior.

c. Cambiar `ASTRA_IMAGE_REGISTRY` para la `acc-operator-controller-manager` imagen a la ruta del registro en la que se insertó la imagen en un paso anterior.

Amplíe el ejemplo astra_control_center_operator_deploy.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: ASTRA_IMAGE_REGISTRY/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_INSTALLTIMEOUT
          value: 5m
        image: ASTRA_IMAGE_REGISTRY/acc-operator:23.10.72
        imagePullPolicy: IfNotPresent
        livenessProbe:
          httpGet:
            path: /healthz
```

```
            port: 8081
          initialDelaySeconds: 15
          periodSeconds: 20
        name: manager
        readinessProbe:
          httpGet:
            path: /readyz
            port: 8081
          initialDelaySeconds: 5
          periodSeconds: 10
        resources:
          limits:
            cpu: 300m
            memory: 750Mi
          requests:
            cpu: 100m
            memory: 75Mi
        securityContext:
          allowPrivilegeEscalation: false
      imagePullSecrets: []
      securityContext:
        runAsUser: 65532
      terminationGracePeriodSeconds: 10
```

3. Instale el operador de Astra Control Center:

```
kubectl apply -f astra_control_center_operator_deploy.yaml
```

Ampliar para respuesta de muestra:

```
namespace/netapp-acc-operator created
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.as
tra.netapp.io created
role.rbac.authorization.k8s.io/acc-operator-leader-election-role
created
clusterrole.rbac.authorization.k8s.io/acc-operator-manager-role
created
clusterrole.rbac.authorization.k8s.io/acc-operator-metrics-reader
created
clusterrole.rbac.authorization.k8s.io/acc-operator-proxy-role
created
rolebinding.rbac.authorization.k8s.io/acc-operator-leader-election-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-manager-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-proxy-
rolebinding created
configmap/acc-operator-manager-config created
service/acc-operator-controller-manager-metrics-service created
deployment.apps/acc-operator-controller-manager created
```

4. Verifique que los pods se estén ejecutando:

```
kubectl get pods -n netapp-acc-operator
```

Configurar Astra Control Center

1. Edite el archivo de recursos personalizados (CR) del Centro de control de Astra (`astra_control_center.yaml`) para realizar las configuraciones de cuenta, soporte, registro y otras necesarias:

```
vim astra_control_center.yaml
```

Una muestra anotada de AYLMA sigue estos pasos.

2. Modifique o confirme los siguientes ajustes:

<code>accountName</code>

Ajuste	Orientación	Tipo	Ejemplo
`accountName`	Cambie el `accountName` Cadena con el nombre que desea asociar a la cuenta Astra Control Center. Sólo puede haber un nombre de cuenta.	cadena	`Example`

<code>astraVersion</code>

Ajuste	Orientación	Tipo	Ejemplo
`astraVersion`	La versión de Astra Control Center para implementar. No se necesita ninguna acción para este ajuste, ya que el valor se rellenará previamente.	cadena	`23.10.0-68`

<code>astraAddress</code>

Ajuste	Orientación	Tipo	Ejemplo
`astraAddress`	Cambie el `astraAddress` Cadena al FQDN (recomendado) o dirección IP que desea utilizar en su navegador para acceder a Astra Control Center. Esta dirección define cómo se encontrará Astra Control Center en su centro de datos y es el mismo FQDN o la dirección IP que ha aprovisionado desde su equilibrador de carga cuando ha finalizado "Requisitos del Centro de Control de Astra". NOTA: No utilizar `http://` o. `https://` en la dirección. Copie este FQDN para utilizarlo en un paso posterior.	cadena	`astra.example.com`

<code>autoSupport</code>

Las selecciones de esta sección determinan si participará en la aplicación de soporte proactivo de NetApp, Active IQ de NetApp y dónde se envían los datos. Se requiere una conexión a Internet (puerto 442) y todos los datos de soporte se anóniman.

Ajuste	Uso	Orientación	Tipo	Ejemplo
`autoSupport.enrolled`	Uno de los dos `enrolled` o. `url` los campos deben seleccionarse	Cambiar `enrolled` Para AutoSupport a. `false` para sitios sin conexión a internet o retención `true` para sitios conectados. Un valor de `true` Permite enviar datos anónimos a NetApp con fines de soporte. La elección predeterminada es `false` E indica que no se enviará ningún dato de soporte a NetApp.	Booleano	`false` (este valor es el predeterminado)
`autoSupport.url`	Uno de los dos `enrolled` o. `url` los campos deben seleccionarse	Esta URL determina dónde se enviarán los datos anónimos.	cadena	`https://support.netapp.com/asupprod/post/1.0/postAsup`

<code>email</code>

Ajuste	Orientación	Tipo	Ejemplo
`email`	Cambie el `email` cadena en la dirección inicial predeterminada del administrador. Copie esta dirección de correo electrónico para su uso en un paso posterior. Esta dirección de correo electrónico se utilizará como nombre de usuario de la cuenta inicial para iniciar sesión en la interfaz de usuario y se le notificarán los eventos de Astra Control.	cadena	`admin@example.com`

<code>firstName</code>

Ajuste	Orientación	Tipo	Ejemplo
`firstName`	El nombre del administrador inicial predeterminado asociado con la cuenta Astra. El nombre utilizado aquí aparecerá en un encabezado de la interfaz de usuario después del primer inicio de sesión.	cadena	`SRE`

<code>LastName</code>

Ajuste	Orientación	Tipo	Ejemplo
`lastName`	Apellido del administrador inicial predeterminado asociado con la cuenta Astra. El nombre utilizado aquí aparecerá en un encabezado de la interfaz de usuario después del primer inicio de sesión.	cadena	`Admin`

<code>imageRegistry</code>

Las selecciones realizadas en esta sección definen el registro de imágenes del contenedor que aloja las imágenes de la aplicación Astra, el operador del centro de control Astra y el repositorio de Astra Control Center Helm.

Ajuste	Uso	Orientación	Tipo	Ejemplo
`imageRegistry.name`	Obligatorio	El nombre del registro de imágenes en el que se insertó las imágenes en el paso anterior. No utilizar `http://` o. `https://` en el nombre del registro.	cadena	`example.registry.com/astra`
`imageRegistry.secret`	Obligatorio si la cadena introducida para `imageRegistry.name' requires a secret.` `IMPORTANT: If you are using a registry that does not require authorization, you must delete this` `secret` línea dentro `imageRegistry` o se producirá un error en la instalación.	El nombre del secreto Kubernetes utilizado para autenticarse con el registro de imágenes.	cadena	`astra-registry-cred`

<code>storageClass</code>

Ajuste	Orientación	Tipo	Ejemplo
`storageClass`	Cambie el `storageClass` valor desde `ontap-gold` A otro recurso de la clase de almacenamiento de Astra Trident, según lo requiera la instalación. Ejecute el comando `kubectl get sc` para determinar las clases de almacenamiento configuradas existentes. Debe introducirse una de las clases de almacenamiento basadas en Astra Trident en el archivo de manifiesto (`astra-control-center-<version>.manifest`) Y se utilizará para Astra PVs. Si no está establecida, se utilizará la clase de almacenamiento predeterminada. NOTA: Si se ha configurado una clase de almacenamiento predeterminada, asegúrese de que es la única clase de almacenamiento que tiene la anotación predeterminada.	cadena	`ontap-gold`

<code>volumeReclaimPolicy</code>

Ajuste	Orientación	Tipo	Opciones
`volumeReclaimPolicy`	De esta forma se establece la política de reclamaciones para los vehículos de Astra. Configuración de esta directiva como `Retain` Conserva los volúmenes persistentes una vez que Astra se elimina. Configuración de esta directiva como `Delete` elimina los volúmenes persistentes después de eliminar astra. Si no se establece este valor, se conservan los VP.	cadena	• `Retain` (Este es el valor predeterminado) • `Delete`

<code>ingressType</code>

Ajuste	Orientación	Tipo	Opciones
`ingressType`	Utilice uno de los siguientes tipos de entrada: `Generic`* (`ingressType: "Generic"`) (Predeterminado) Utilice esta opción cuando tenga otro controlador de entrada en uso o prefiera utilizar su propio controlador de entrada. Después de implementar Astra Control Center, deberá configurar el "controlador de entrada" Para exponer Astra Control Center con una URL. IMPORTANTE: Si va a utilizar una malla de servicio con Astra Control Center, debe seleccionar `Generic` como tipo de ingreso y configure el suyo propio "controlador de entrada". **`AccTraefik`** (`ingressType: "AccTraefik"`) Utilice esta opción cuando prefiera no configurar un controlador de entrada. Esto despliega el Astra Control Center `traefik` Puerta de enlace como servicio de tipo Kubernetes LoadBalancer. Astra Control Center utiliza un servicio del tipo "LoadBalancer" (`svc/traefik` En el espacio de nombres de Astra Control Center) y requiere que se le	cadena	• `Generic` (este es el valor predeterminado) • `AccTraefik`

<code>scaleSize</code>

Ajuste	Orientación	Tipo	Opciones
`scaleSize`	De forma predeterminada, Astra utilizará la alta disponibilidad (HA) `scaleSize` de `Medium`, Que despliega la mayoría de los servicios en HA y despliega múltiples réplicas para redundancia. Con `scaleSize` como `Small`, Astra reducirá el número de réplicas para todos los servicios excepto los servicios esenciales para reducir el consumo. CONSEJO: `Medium` las puestas en marcha constan de unos 100 pods (sin incluir cargas de trabajo transitorias. 100 pod se basa en la configuración de tres nodos principales y tres nodos de trabajador). Tenga en cuenta las limitaciones de límites de red por pod que pueden ser un problema en su entorno, sobre todo cuando tenga en cuenta situaciones de recuperación ante desastres.	cadena	• `Small` • `Medium` (Este es el valor predeterminado)

<code>astraResourcesScaler</code>

Ajuste	Orientación	Tipo	Opciones
`astraResourcesScaler`	Opciones de escalado para los límites de recursos de AstraControlCenter. De forma predeterminada, Astra Control Center se despliega con solicitudes de recursos establecidas para la mayoría de los componentes de Astra. Esta configuración permite que la pila de software de Astra Control Center tenga un mejor rendimiento en entornos con un mayor nivel de carga y escalabilidad de las aplicaciones. Sin embargo, en situaciones que utilizan grupos de desarrollo o pruebas más pequeños, el campo CR `astraResourcesScalar` se puede establecer en `Off`. De este modo se deshabilitan las solicitudes de recursos y se puede implementar en clústeres más pequeños.	cadena	• `Default` (Este es el valor predeterminado) • `Off`

<code>additionalValues</code>

Añada los siguientes valores adicionales a Astra Control Center CR para evitar un problema conocido en la instalación:

```
additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
```

- Para el Centro de control astral y la comunicación Cloud Insights, la verificación de certificados TLS está desactivada de forma predeterminada. Puede habilitar la verificación de la certificación TLS para la comunicación entre Cloud Insights y el clúster de host del Centro de control de Astra y el clúster gestionado, añadiendo la siguiente sección en la `additionalValues`.

```
 additionalValues:
   netapp-monitoring-operator:
     config:
       ciSkipTlsVerify: false
   cloud-insights-service:
     config:
       ciSkipTlsVerify: false
   telemetry-service:
     config:
       ciSkipTlsVerify: false
```

<code>crds</code>

Sus selecciones en esta sección determinan cómo Astra Control Center debe manejar los CRD.

Ajuste	Orientación	Tipo	Ejemplo
`crds.externalCertManager`	Si utiliza un administrador de certificados externo, cambie `externalCertManager` para `true`. El valor predeterminado `false` Hace que Astra Control Center instale sus propios CRD de administrador de certificados durante la instalación. Los crds son objetos de todo el clúster y su instalación podría tener un impacto en otras partes del clúster. Puede utilizar este indicador para indicar a Astra Control Center que el administrador del clúster instalará y gestionará estos CRD fuera de Astra Control Center.	Booleano	`False` (este valor es el predeterminado)
`crds.externalTraefik`	De forma predeterminada, Astra Control Center instalará los CRD de Traefik necesarios. Los crds son objetos de todo el clúster y su instalación podría tener un impacto en otras partes del clúster. Puede utilizar este indicador para indicar a Astra Control Center que el administrador del clúster instalará y gestionará estos CRD fuera de Astra Control Center.	Booleano	`False` (este valor es el predeterminado)

Asegúrese de haber seleccionado la clase de almacenamiento y el tipo de entrada correctos para la configuración antes de completar la instalación.

Expanda para la muestra astra_control_center.yaml

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  scaleSize: "Medium"
  astraResourcesScaler: "Default"
  additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
  crds:
    externalTraefik: false
    externalCertManager: false
```

Complete la instalación del centro de control de Astra y del operador

1. Si todavía no lo ha hecho en un paso anterior, cree el `netapp-acc` espacio de nombres (o personalizado):

```
kubectl create ns [netapp-acc or custom namespace]
```

2. Si usas una malla de servicio con Astra Control Center, agrega la siguiente etiqueta a la `netapp-acc` o espacio de nombres personalizado:

   Su tipo de ingreso (`ingressType`) debe establecerse en `Generic` En Astra Control Center CR antes de continuar con este comando.

   ```
   kubectl label ns [netapp-acc or custom namespace] istio-
   injection:enabled
   ```

3. (Recomendado) "Activar MTLS estricto" Para la malla de servicio de Istio:

   ```
   kubectl apply -n istio-system -f - <<EOF
   apiVersion: security.istio.io/v1beta1
   kind: PeerAuthentication
   metadata:
     name: default
   spec:
     mtls:
       mode: STRICT
   EOF
   ```

4. Instale Astra Control Center en `netapp-acc` (o su espacio de nombres personalizado):

   ```
   kubectl apply -f astra_control_center.yaml -n [netapp-acc or custom
   namespace]
   ```

   El operador del Centro de control de Astra realizará una comprobación automática de los requisitos del entorno. Ausente "requisitos" Puede provocar que falle la instalación o que Astra Control Center no funcione correctamente. Consulte siguiente sección para comprobar si hay mensajes de advertencia relacionados con la comprobación automática del sistema.

Comprobar el estado del sistema

Puede verificar el estado del sistema con comandos kubectl. Si prefiere utilizar OpenShift, puede utilizar comandos de OC comparables para realizar los pasos de verificación.

Pasos

1. Compruebe que el proceso de instalación no ha generado mensajes de advertencia relacionados con las comprobaciones de validación:

   ```
   kubectl get acc [astra or custom Astra Control Center CR name] -n
   [netapp-acc or custom namespace] -o yaml
   ```

También se notifican mensajes de advertencia adicionales en los registros del operador de Astra Control Center.

2. Corrija cualquier problema del entorno que se notifique mediante las comprobaciones automatizadas de requisitos.

Puede corregir problemas garantizando que su entorno cumple con los "requisitos" Para Astra Control Center.

3. Compruebe que todos los componentes del sistema se han instalado correctamente.

```
kubectl get pods -n [netapp-acc or custom namespace]
```

Cada pod debe tener el estado de `Running`. Pueden tardar varios minutos en implementar los pods del sistema.

Amplíe para obtener una respuesta de muestra

```
NAME                                          READY   STATUS
RESTARTS     AGE
acc-helm-repo-6cc7696d8f-pmhm8                1/1     Running     0
9h
activity-597fb656dc-5rd4l                     1/1     Running     0
9h
activity-597fb656dc-mqmcw                     1/1     Running     0
9h
api-token-authentication-62f84                1/1     Running     0
9h
api-token-authentication-68nlf                1/1     Running     0
9h
api-token-authentication-ztgrm                1/1     Running     0
9h
asup-669d4ddbc4-fnmwp                         1/1     Running     1
(9h ago)   9h
authentication-78789d7549-lk686               1/1     Running     0
9h
bucketservice-65c7d95496-24x7l                1/1     Running     3
(9h ago)   9h
cert-manager-c9f9fbf9f-k8zq2                  1/1     Running     0
9h
cert-manager-c9f9fbf9f-qjlzm                  1/1     Running     0
9h
cert-manager-cainjector-dbbbd8447-b5qll       1/1     Running     0
9h
cert-manager-cainjector-dbbbd8447-p5whs       1/1     Running     0
9h
cert-manager-webhook-6f97bb7d84-4722b         1/1     Running     0
9h
cert-manager-webhook-6f97bb7d84-86kv5         1/1     Running     0
9h
certificates-59d9f6f4bd-2j899                 1/1     Running     0
9h
certificates-59d9f6f4bd-9d9k6                 1/1     Running     0
9h
certificates-expiry-check-28011180--1-8lkxz   0/1     Completed   0
9h
cloud-extension-5c9c9958f8-jdhrp              1/1     Running     0
9h
cloud-insights-service-5cdd5f7f-pp8r5         1/1     Running     0
9h
composite-compute-66585789f4-hxn5w            1/1     Running     0
9h
```

```
composite-volume-68649f68fd-tb7p4             1/1     Running     0
9h
credentials-dfc844c57-jsx92                   1/1     Running     0
9h
credentials-dfc844c57-xw26s                   1/1     Running     0
9h
entitlement-7b47769b87-4jb6c                  1/1     Running     0
9h
features-854d8444cc-c24b7                     1/1     Running     0
9h
features-854d8444cc-dv6sm                     1/1     Running     0
9h
fluent-bit-ds-9tlv4                           1/1     Running     0
9h
fluent-bit-ds-bpkcb                           1/1     Running     0
9h
fluent-bit-ds-cxmwx                           1/1     Running     0
9h
fluent-bit-ds-jgnhc                           1/1     Running     0
9h
fluent-bit-ds-vtr6k                           1/1     Running     0
9h
fluent-bit-ds-vxqd5                           1/1     Running     0
9h
graphql-server-7d4b9d44d5-zdbf5               1/1     Running     0
9h
identity-6655c48769-4pwk8                     1/1     Running     0
9h
influxdb2-0                                   1/1     Running     0
9h
keycloak-operator-55479d6fc6-slvmt            1/1     Running     0
9h
krakend-f487cb465-78679                       1/1     Running     0
9h
krakend-f487cb465-rjsxx                       1/1     Running     0
9h
license-64cbc7cd9c-qxsr8                      1/1     Running     0
9h
login-ui-5db89b5589-ndb96                     1/1     Running     0
9h
loki-0                                        1/1     Running     0
9h
metrics-facade-8446f64c94-x8h7b               1/1     Running     0
9h
monitoring-operator-6b44586965-pvcl4          2/2     Running     0
9h
```

```
nats-0                                          1/1     Running     0
9h
nats-1                                          1/1     Running     0
9h
nats-2                                          1/1     Running     0
9h
nautilus-85754d87d7-756qb                       1/1     Running     0
9h
nautilus-85754d87d7-q8j7d                       1/1     Running     0
9h
openapi-5f9cc76544-7fnjm                        1/1     Running     0
9h
openapi-5f9cc76544-vzr7b                        1/1     Running     0
9h
packages-5db49f8b5-lrzhd                        1/1     Running     0
9h
polaris-consul-consul-server-0                  1/1     Running     0
9h
polaris-consul-consul-server-1                  1/1     Running     0
9h
polaris-consul-consul-server-2                  1/1     Running     0
9h
polaris-keycloak-0                              1/1     Running     2
(9h ago)   9h
polaris-keycloak-1                              1/1     Running     0
9h
polaris-keycloak-2                              1/1     Running     0
9h
polaris-keycloak-db-0                           1/1     Running     0
9h
polaris-keycloak-db-1                           1/1     Running     0
9h
polaris-keycloak-db-2                           1/1     Running     0
9h
polaris-mongodb-0                               1/1     Running     0
9h
polaris-mongodb-1                               1/1     Running     0
9h
polaris-mongodb-2                               1/1     Running     0
9h
polaris-ui-66fb99479-qp9gq                      1/1     Running     0
9h
polaris-vault-0                                 1/1     Running     0
9h
polaris-vault-1                                 1/1     Running     0
9h
```

```
polaris-vault-2                                1/1     Running     0
9h
public-metrics-76fbf9594d-zmxzw                1/1     Running     0
9h
storage-backend-metrics-7d7fbc9cb9-lmd25       1/1     Running     0
9h
storage-provider-5bdd456c4b-2fftc              1/1     Running     0
9h
task-service-87575df85-dnn2q                   1/1     Running     3
(9h ago)   9h
task-service-task-purge-28011720--1-q6w4r      0/1     Completed   0
28m
task-service-task-purge-28011735--1-vk6pd      1/1     Running     0
13m
telegraf-ds-2r2kw                              1/1     Running     0
9h
telegraf-ds-6s9d5                              1/1     Running     0
9h
telegraf-ds-96jl7                              1/1     Running     0
9h
telegraf-ds-hbp84                              1/1     Running     0
9h
telegraf-ds-plwzv                              1/1     Running     0
9h
telegraf-ds-sr22c                              1/1     Running     0
9h
telegraf-rs-4sbg8                              1/1     Running     0
9h
telemetry-service-fb9559f7b-mk9l7              1/1     Running     3
(9h ago)   9h
tenancy-559bbc6b48-5msgg                       1/1     Running     0
9h
traefik-d997b8877-7xpf4                        1/1     Running     0
9h
traefik-d997b8877-9xv96                        1/1     Running     0
9h
trident-svc-585c97548c-d25z5                   1/1     Running     0
9h
vault-controller-88484b454-2d6sr               1/1     Running     0
9h
vault-controller-88484b454-fc5cz               1/1     Running     0
9h
vault-controller-88484b454-jktld               1/1     Running     0
9h
```

4. (Opcional) Vea el `acc-operator` registros para supervisar el progreso:

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

> `accHost` el registro del clúster es una de las últimas operaciones y, si no se produce un error en la implementación, no provocará un error. En el caso de que se produzca un error de registro del clúster que se indica en los registros, puede volver a intentar realizar el registro a través de la "Añada el flujo de trabajo del clúster en la interfaz de usuario de" O API.

5. Cuando todos los pods estén en ejecución, verifique que la instalación se haya realizado correctamente (`READY` es `True`) Y obtenga la contraseña de configuración inicial que utilizará cuando inicie sesión en Astra Control Center:

```
kubectl get AstraControlCenter -n [netapp-acc or custom namespace]
```

Respuesta:

```
NAME    UUID                                   VERSION     ADDRESS
READY
astra   9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f  23.10.0-68
10.111.111.111  True
```

> Copie el valor de UUID. La contraseña es `ACC-` Seguido del valor UUID (`ACC-[UUID]` o, en este ejemplo, `ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f`).

Configure la entrada para el equilibrio de carga

Puede configurar un controlador de entrada de Kubernetes que gestione el acceso externo a los servicios. Estos procedimientos proporcionan ejemplos de configuración para un controlador de entrada si utilizó el valor predeterminado de `ingressType: "Generic"` En el recurso personalizado Astra Control Center (`astra_control_center.yaml`). No es necesario utilizar este procedimiento si se ha especificado `ingressType: "AccTraefik"` En el recurso personalizado Astra Control Center (`astra_control_center.yaml`).

Después de implementar Astra Control Center, deberá configurar el controlador Ingress para exponer Astra Control Center con una dirección URL.

Los pasos de configuración varían en función del tipo de controlador de entrada que utilice. Astra Control Center admite muchos tipos de controladores Ingress. Estos procedimientos de configuración proporcionan pasos de ejemplo para algunos tipos de controladores de entrada comunes.

Antes de empezar

- El requerido "controlador de entrada" ya debe ponerse en marcha.

- La "clase de entrada" ya se debe crear la correspondiente al controlador de entrada.

Pasos para la entrada de Istio

1. Configurar la entrada de Istio.

   En este procedimiento se asume que Istio se implementa utilizando el perfil de configuración "predeterminado".

2. Recopile o cree el certificado y el archivo de claves privadas deseados para la puerta de enlace de entrada.

   Es posible usar un certificado firmado por CA o autofirmado. El nombre común debe ser la dirección Astra (FQDN).

   Comando de ejemplo:

   ```
   openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key
   -out tls.crt
   ```

3. Cree un secreto `tls secret name` de tipo `kubernetes.io/tls` Para una clave privada TLS y un certificado en `istio-system namespace` Tal como se describe en los secretos TLS.

   Comando de ejemplo:

   ```
   kubectl create secret tls [tls secret name] --key="tls.key"
   --cert="tls.crt" -n istio-system
   ```

   El nombre del secreto debe coincidir con el `spec.tls.secretName` proporcionado en `istio-ingress.yaml` archivo.

4. Implemente un recurso de entrada en `netapp-acc` espacio de nombres (o con nombre personalizado) mediante el tipo de recurso v1 para un esquema (`istio-Ingress.yaml` se utiliza en este ejemplo):

```
apiVersion: networking.k8s.io/v1
kind: IngressClass
metadata:
  name: istio
spec:
  controller: istio.io/ingress-controller
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: istio
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: traefik
            port:
              number: 80
```

5. Aplicar los cambios:

```
kubectl apply -f istio-Ingress.yaml
```

6. Compruebe el estado de la entrada:

```
kubectl get ingress -n [netapp-acc or custom namespace]
```

Respuesta:

```
NAME    CLASS HOSTS             ADDRESS         PORTS   AGE
ingress istio astra.example.com 172.16.103.248  80, 443 1h
```

7. Finalice la instalación de Astra Control Center.

Pasos para el controlador de entrada Nginx

1. Cree un secreto de tipo `kubernetes.io/tls` Para una clave privada TLS y un certificado en `netapp-acc` (o nombre personalizado) como se describe en "Secretos TLS".
2. Implemente un recurso de entrada en `netapp-acc` espacio de nombres (o con nombre personalizado) mediante el tipo de recurso v1 para un esquema (`nginx-Ingress.yaml` se utiliza en este ejemplo):

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netapp-acc-ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: [class name for nginx controller]
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: <ACC address>
    http:
      paths:
        - path:
          backend:
            service:
              name: traefik
              port:
                number: 80
          pathType: ImplementationSpecific
```

3. Aplicar los cambios:

```
kubectl apply -f nginx-Ingress.yaml
```

NetApp recomienda la instalación de la controladora nginx como una puesta en marcha en lugar de como una `daemonSet`.

Pasos para el controlador de entrada de OpenShift

1. Obtenga su certificado y consiga los archivos de clave, certificado y CA listos para su uso por la ruta OpenShift.
2. Cree la ruta OpenShift:

```
oc create route edge --service=traefik --port=web -n [netapp-acc or
custom namespace] --insecure-policy=Redirect --hostname=<ACC
address> --cert=cert.pem --key=key.pem
```

Inicie sesión en la interfaz de usuario de Astra Control Center

Después de instalar Astra Control Center, cambiará la contraseña del administrador predeterminado e inicie sesión en el panel de interfaz de usuario de Astra Control Center.

Pasos

1. En un navegador, introduzca el FQDN (incluido el `https://` prefijo) que utilizó en el `astraAddress` en la `astra_control_center.yaml` CR cuando Ha instalado Astra Control Center.
2. Acepte los certificados autofirmados si se le solicita.

   Se puede crear un certificado personalizado después de iniciar sesión.

3. En la página de inicio de sesión de Astra Control Center, introduzca el valor utilizado `email` pulg `astra_control_center.yaml` CR cuando Ha instalado Astra Control Center, seguido de la contraseña de configuración inicial (`ACC-[UUID]`).

   Si introduce una contraseña incorrecta tres veces, la cuenta de administrador se bloqueará durante 15 minutos.

4. Seleccione **Iniciar sesión**.
5. Cambie la contraseña cuando se le solicite.

   Si éste es su primer inicio de sesión y olvida la contraseña y no se han creado otras cuentas de usuario administrativas, póngase en contacto con "Soporte de NetApp" para obtener ayuda para la recuperación de contraseñas.

6. (Opcional) quite el certificado TLS autofirmado existente y sustitúyalo por un "Certificado TLS personalizado firmado por una entidad de certificación (CA)".

Solucione los problemas de instalación

Si alguno de los servicios está en `Error` puede inspeccionar los registros. Busque códigos de respuesta API en la gama 400 a 500. Esos indican el lugar donde ocurrió un fracaso.

Opciones

- Para inspeccionar los registros del operador de Astra Control Center, introduzca lo siguiente:

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

- Para comprobar el resultado de Astra Control Center CR:

```
kubectl get acc -n [netapp-acc or custom namespace] -o yaml
```

El futuro

- (Opcional) en función de su entorno, post-instalación completa "pasos de configuración".
- Complete la implementación llevando a cabo "tareas de configuración".

Configure un administrador de certificados externo

Si ya existe un administrador de certificados en su clúster de Kubernetes, deberá realizar algunos pasos previos para que Astra Control Center no instale su propio administrador de certificados.

Pasos

1. Confirme que tiene instalado un administrador de certificados:

```
kubectl get pods -A | grep 'cert-manager'
```

Respuesta de ejemplo:

```
cert-manager   essential-cert-manager-84446f49d5-sf2zd              1/1
Running    0      6d5h
cert-manager   essential-cert-manager-cainjector-66dc99cc56-9ldmt   1/1
Running    0      6d5h
cert-manager   essential-cert-manager-webhook-56b76db9cc-fjqrq      1/1
Running    0      6d5h
```

2. Cree un certificado/pareja de claves para `astraAddress` FQDN:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out
tls.crt
```

Respuesta de ejemplo:

```
Generating a 2048 bit RSA private key
...................+++
.......................................+++
writing new private key to 'tls.key'
```

3. Crear un secreto con archivos generados previamente:

```
kubectl create secret tls selfsigned-tls --key tls.key --cert tls.crt -n
<cert-manager-namespace>
```

Respuesta de ejemplo:

```
secret/selfsigned-tls created
```

4. Cree un `ClusterIssuer` archivo que es **exactamente** el siguiente pero que incluye la ubicación del espacio de nombres donde el `cert-manager` los pods están instalados:

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: astra-ca-clusterissuer
  namespace: <cert-manager-namespace>
spec:
  ca:
    secretName: selfsigned-tls
```

```
kubectl apply -f ClusterIssuer.yaml
```

Respuesta de ejemplo:

```
clusterissuer.cert-manager.io/astra-ca-clusterissuer created
```

5. Compruebe que el `ClusterIssuer` ha surgido correctamente. `Ready` debe ser `True` antes de poder continuar:

```
kubectl get ClusterIssuer
```

Respuesta de ejemplo:

```
NAME                        READY   AGE
astra-ca-clusterissuer      True    9s
```

6. Complete el "Proceso de instalación de Astra Control Center". Hay una "Paso de configuración necesario para el clúster YAML de Astra Control Center" En el que cambia el valor CRD para indicar que el administrador de certificados está instalado externamente. Debe completar este paso durante la instalación para que Astra Control Center reconozca al gestor de certificados externo.

Instale Astra Control Center utilizando OpenShift OperatorHub

Si utiliza Red Hat OpenShift, puede instalar Astra Control Center mediante el operador certificado de Red Hat. Utilice este procedimiento para instalar Astra Control Center desde "Catálogo de Red Hat Ecosystem" O con Red Hat OpenShift Container Platform.

Después de completar este procedimiento, debe volver al procedimiento de instalación para completar el "pasos restantes" para verificar que la instalación se ha realizado correctamente e iniciar sesión.

Antes de empezar

- **Cumplir con los requisitos ambientales**: "Antes de comenzar la instalación, prepare su entorno para la implementación de Astra Control Center".
- * Asegurar operadores de clúster saludables y servicios API*:
  - En el clúster de OpenShift, asegúrese de que todos los operadores de clúster se encuentran en buen estado:

    ```
    oc get clusteroperators
    ```

  - En el clúster OpenShift, asegúrese de que todos los servicios API se encuentran en buen estado:

    ```
    oc get apiservices
    ```

- **Asegúrese de que un FQDN enrutable**: El FQDN de Astra que planea utilizar se puede enrutar al clúster. Esto significa que tiene una entrada DNS en el servidor DNS interno o que está utilizando una ruta URL principal que ya está registrada.
- **Obtenga permisos OpenShift**: Necesitará todos los permisos necesarios y acceso a Red Hat OpenShift Container Platform para realizar los pasos de instalación descritos.
- **Configurar un administrador de cert**: Si ya existe un administrador de cert en el clúster, debe realizar algunos "requisitos previos" Por lo tanto, Astra Control Center no instala su propio administrador de certificados. De forma predeterminada, Astra Control Center instala su propio administrador de certificados durante la instalación.
- **Considera una malla de servicio**: Se recomienda encarecidamente que los canales de comunicaciones del clúster host de Astra Control estén protegidos mediante un "malla de servicio compatible".

Detalles de malla de servicio de Istio

Para el uso de la malla de servicio de Istio, deberá hacer lo siguiente:

- Agregue un `istio-injection:enabled` Etiqueta en el espacio de nombres de Astra antes de implementar Astra Control Center.
- Utilice la `Generic` ajuste de entrada y proporcionar una entrada alternativa para "equilibrio de carga externo".
- Para los clústeres de Red Hat OpenShift, deberá definirlos `NetworkAttachmentDefinition` En todos los espacios de nombres del Centro de control de Astra asociados (`netapp-acc-operator`, `netapp-acc`, `netapp-monitoring` para clusters de aplicaciones o cualquier espacio de nombres personalizado que se haya sustituido).

```
cat <<EOF | oc -n netapp-acc-operator create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-acc create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-monitoring create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF
```

- **Controlador de entrada de Kubernetes**: Si tiene un controlador de entrada de Kubernetes que gestiona el acceso externo a servicios, como el equilibrio de carga en un clúster, debe configurarlo para su uso con Astra Control Center:

a. Crear el espacio de nombres del operador:

```
oc create namespace netapp-acc-operator
```

b. "Completar la configuración" para el tipo de controlador de entrada.

- **Solo controlador SAN de ONTAP**: Si está utilizando un controlador SAN de ONTAP, asegúrese de que la función multivía esté habilitada en todos sus clústeres de Kubernetes.

Pasos

- Descargue y extraiga Astra Control Center
- Instale el complemento Astra kubectl de NetApp
- Agregue las imágenes al registro local
- Busque la página de instalación del operador
- Instale el operador
- Instalar Astra Control Center

Descargue y extraiga Astra Control Center

Puede elegir descargar el paquete Astra Control Center desde el sitio de soporte de NetApp o utilizar Docker para extraer el paquete del registro de imágenes del servicio de control de Astra.

Sitio de soporte de NetApp

1. Descargue el paquete que contiene Astra Control Center (`astra-control-center-[version].tar.gz`) del "Página de descargas de Astra Control Center".
2. (Recomendado pero opcional) Descargue el paquete de certificados y firmas para Astra Control Center (`astra-control-center-certs-[version].tar.gz`) para verificar la firma del paquete.

   Amplíe para obtener más detalles

   ```
   tar -vxzf astra-control-center-certs-[version].tar.gz
   ```

   ```
   openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub
   -signature certs/astra-control-center-[version].tar.gz.sig
   astra-control-center-[version].tar.gz
   ```

   Se mostrará la salida `Verified OK` después de una verificación correcta.

3. Extraiga las imágenes del paquete Astra Control Center:

   ```
   tar -vxzf astra-control-center-[version].tar.gz
   ```

Registro de imágenes de Astra Control

1. Inicia sesión en el servicio Astra Control.
2. En el Dashboard, selecciona **Desplegar una instancia autogestionada de Astra Control**.
3. Sigue las instrucciones para iniciar sesión en el registro de imágenes de Astra Control, extraer la imagen de instalación de Astra Control Center y extraer la imagen.

Instale el complemento Astra kubectl de NetApp

Puede utilizar el complemento de línea de comandos kubectl de Astra de NetApp para insertar imágenes en un repositorio de Docker local.

Antes de empezar

NetApp proporciona binarios de complementos para diferentes arquitecturas de CPU y sistemas operativos. Debe saber qué CPU y sistema operativo tiene antes de realizar esta tarea.

Pasos

1. Enumere los binarios disponibles del complemento Astra kubectl de NetApp, y anote el nombre del archivo que necesita para el sistema operativo y la arquitectura de CPU:

   La biblioteca de complementos kubectl forma parte del paquete tar y se extrae en la carpeta `kubectl-astra`.

```
ls kubectl-astra/
```

2. Mueva el binario correcto a la ruta actual y cambie el nombre a. `kubectl-astra`:

```
cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
```

Agregue las imágenes al registro local

1. Complete la secuencia de pasos apropiada para el motor del contenedor:

Docker

1. Cambie al directorio raíz del tarball. Debería ver el `acc.manifest.bundle.yaml` archivo y estos directorios:

   ```
   acc/
   kubectl-astra/
   acc.manifest.bundle.yaml
   ```

2. Inserte las imágenes del paquete en el directorio de imágenes de Astra Control Center en su registro local. Realice las siguientes sustituciones antes de ejecutar el `push-images` comando:

   - Sustituya <BUNDLE_FILE> por el nombre del archivo Astra Control Bundle (`acc.manifest.bundle.yaml`).
   - Sustituya <MY_FULL_REGISTRY_PATH> por la URL del repositorio de Docker; por ejemplo, "<a href="https://<docker-registry>"" class="bare">https://<docker-registry>".
   - Reemplace <MY_REGISTRY_USER> por el nombre de usuario.
   - Sustituya <MY_REGISTRY_TOKEN> por un token autorizado para el registro.

     ```
     kubectl astra packages push-images -m <BUNDLE_FILE> -r
     <MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
     <MY_REGISTRY_TOKEN>
     ```

Podman

1. Cambie al directorio raíz del tarball. Debería ver este archivo y directorio:

   ```
   acc/
   kubectl-astra/
   acc.manifest.bundle.yaml
   ```

2. Inicie sesión en su registro:

   ```
   podman login <YOUR_REGISTRY>
   ```

3. Prepare y ejecute una de las siguientes secuencias de comandos personalizadas para la versión de Podman que utilice. Sustituya <MY_FULL_REGISTRY_PATH> por la URL del repositorio que incluye cualquier subdirectorio.

   ```
   <strong>Podman 4</strong>
   ```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.10.0-68
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/${
PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.10.0-68
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/${
PACKAGEVERSION}/${astraImageNoPath}
done
```

La ruta de acceso de imagen que crea el script debe parecerse a la siguiente, dependiendo de la configuración del Registro:

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/23.10.0-68/image:version
```

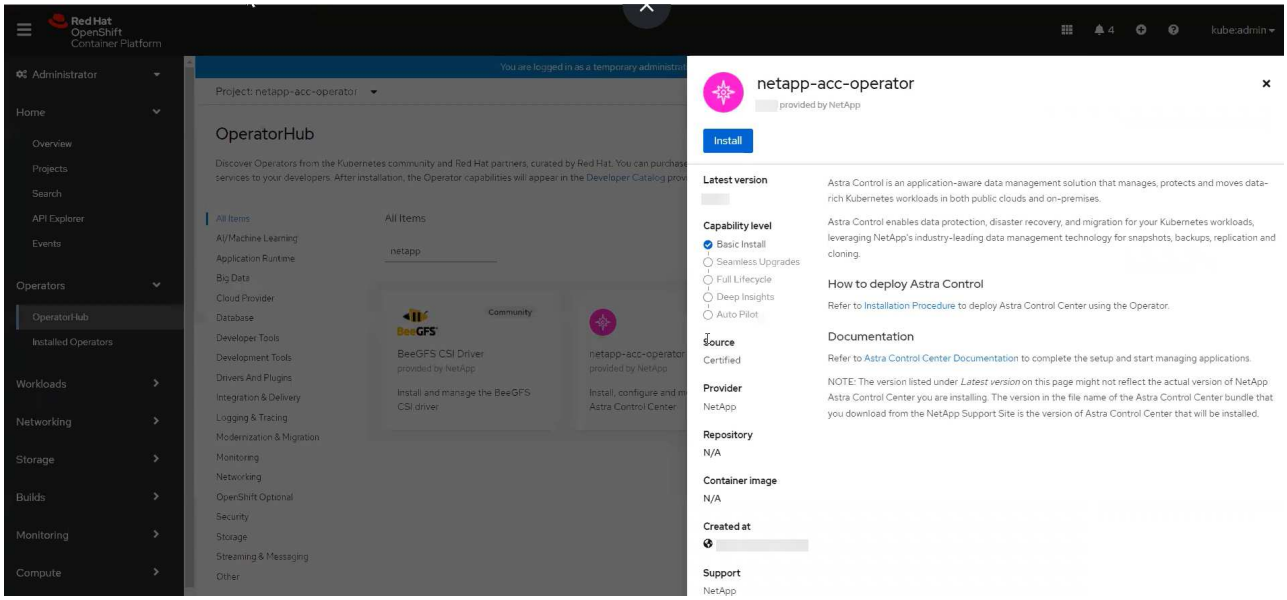
Busque la página de instalación del operador

1. Realice uno de los siguientes procedimientos para acceder a la página de instalación del operador:

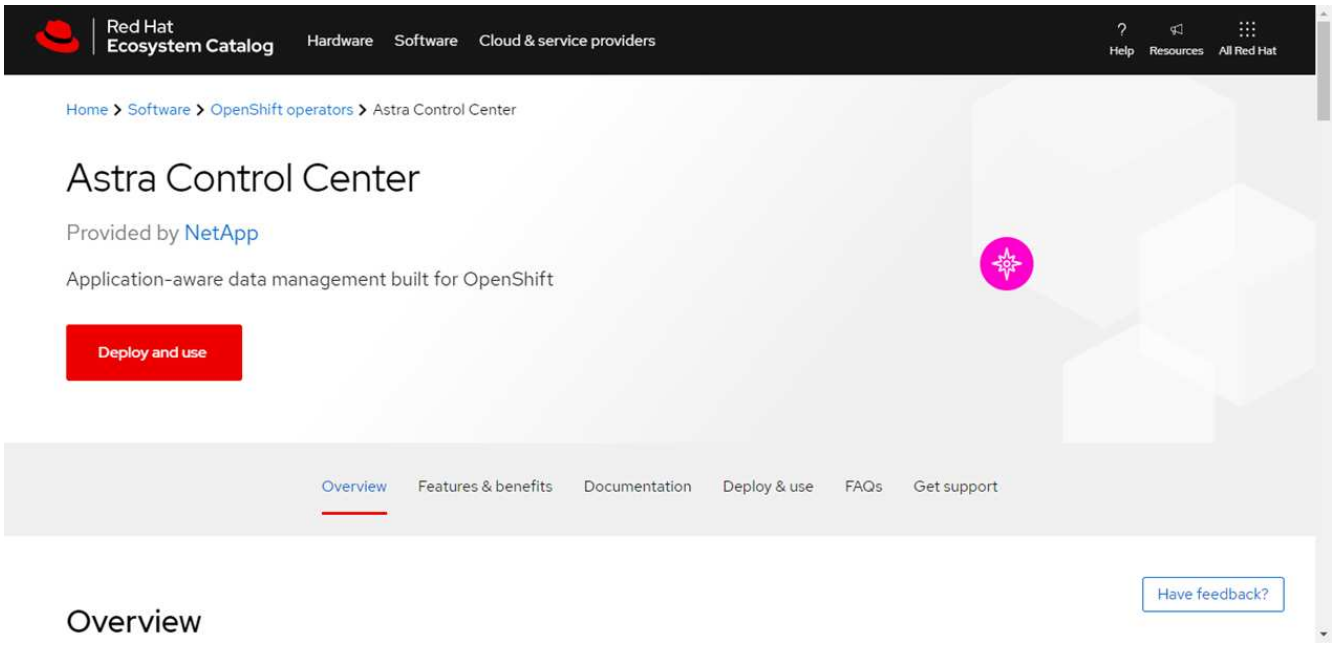
- Desde la consola web de Red Hat OpenShift:
  i. Inicie sesión en la IU de OpenShift Container Platform.
  ii. En el menú lateral, seleccione **operadores > OperatorHub**.

> Solo se puede actualizar a la versión actual de Astra Control Center con este operador.

  iii. Busque y seleccione el operador Centro de control Astra de NetApp.



- En el catálogo de ecosistemas de Red Hat:
  i. Seleccione Astra Control Center de NetApp "operador".
  ii. Seleccione **desplegar y utilizar**.



Instale el operador

1. Complete la página **Install Operator** e instale el operador:

   El operador estará disponible en todos los espacios de nombres del clúster.

   a. Seleccione el espacio de nombres del operador o. `netapp-acc-operator` el espacio de nombres se creará automáticamente como parte de la instalación del operador.

   b. Seleccione una estrategia de aprobación manual o automática.

      Se recomienda la aprobación manual. Solo debe tener una instancia de operador en ejecución por clúster.

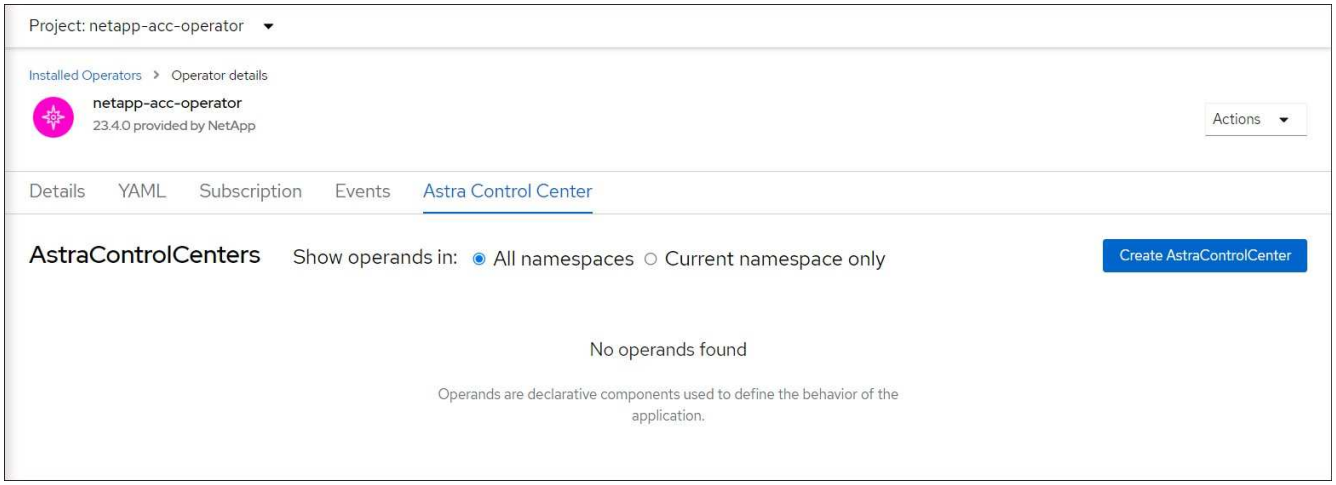
   c. Seleccione **instalar**.

      Si ha seleccionado una estrategia de aprobación manual, se le pedirá que apruebe el plan de instalación manual para este operador.

2. Desde la consola, vaya al menú OperatorHub y confirme que el operador se ha instalado correctamente.

Instalar Astra Control Center

1. Desde la consola de la pestaña **Astra Control Center** del operador Astra Control Center, seleccione **Crear AstraControlCenter**.



2. Complete el `Create AstraControlCenter` campo de formulario:

   a. Mantenga o ajuste el nombre del Centro de control de Astra.

   b. Agregue etiquetas para Astra Control Center.

   c. Habilitar o deshabilitar AutoSupport. Se recomienda conservar la funcionalidad de AutoSupport.

   d. Introduzca el FQDN o la dirección IP de Astra Control Center. No entre `http://` o. `https://` en el campo de dirección.

   e. Introduce la versión de Astra Control Center; por ejemplo, 23.10.0-68.

   f. Introduzca un nombre de cuenta, una dirección de correo electrónico y un apellido de administrador.

   g. Seleccione una política de reclamaciones de volumen de `Retain`, `Recycle`, o. `Delete`. El valor predeterminado es `Retain`.

h. Seleccione el scaleSize de la instalación.

> De forma predeterminada, Astra utilizará la alta disponibilidad (HA) `scaleSize` de `Medium`, Que despliega la mayoría de los servicios en HA y despliega múltiples réplicas para redundancia. Con `scaleSize` como `Small`, Astra reducirá el número de réplicas para todos los servicios excepto los servicios esenciales para reducir el consumo.

i. Seleccione el tipo de entrada:

- **`Generic`** (`ingressType: "Generic"`) (Predeterminado)

  Utilice esta opción cuando tenga otro controlador de entrada en uso o prefiera utilizar su propio controlador de entrada. Después de implementar Astra Control Center, deberá configurar el "controlador de entrada" Para exponer Astra Control Center con una URL.

- **`AccTraefik`** (`ingressType: "AccTraefik"`)

  Utilice esta opción cuando prefiera no configurar un controlador de entrada. Esto despliega el Astra Control Center `traefik` Puerta de enlace como servicio de tipo "LoadBalancer" de Kubernetes.

Astra Control Center utiliza un servicio del tipo "LoadBalancer" (`svc/traefik` En el espacio de nombres de Astra Control Center) y requiere que se le asigne una dirección IP externa accesible. Si se permiten equilibradores de carga en su entorno y aún no tiene uno configurado, puede utilizar MetalLB u otro equilibrador de carga de servicio externo para asignar una dirección IP externa al servicio. En la configuración interna del servidor DNS, debe apuntar el nombre DNS elegido para Astra Control Center a la dirección IP con equilibrio de carga.

> Para obtener detalles sobre el tipo de servicio de “LoadBalancer” e Ingress, consulte "Requisitos".

a. En **Registro de imágenes**, introduzca la ruta de registro de la imagen del contenedor local. No entre `http://` o. `https://` en el campo de dirección.

b. Si utiliza un registro de imágenes que requiere autenticación, introduzca el secreto de imagen.

> Si utiliza un registro que requiere autenticación, cree un secreto en el clúster.

c. Introduzca el nombre del administrador.

d. Configure el escalado de recursos.

e. Proporcione la clase de almacenamiento predeterminada.

> Si se configura una clase de almacenamiento predeterminada, asegúrese de que es la única clase de almacenamiento que tiene la anotación predeterminada.

f. Defina las preferencias de manejo de CRD.

3. Seleccione la vista YAML para revisar los ajustes seleccionados.
4. Seleccione `Create`.

Cree un secreto de registro

Si utiliza un registro que requiere autenticación, cree un secreto en el clúster de OpenShift e introduzca el nombre secreto en el `Create AstraControlCenter` campo de formulario.

1. Cree un espacio de nombres para el operador Astra Control Center:

```
oc create ns [netapp-acc-operator or custom namespace]
```

2. Cree un secreto en este espacio de nombres:

```
oc create secret docker-registry astra-registry-cred n [netapp-acc-
operator or custom namespace] --docker-server=[your_registry_path]
--docker username=[username] --docker-password=[token]
```

> Astra Control sólo admite secretos de registro Docker.

3. Complete los campos restantes en El campo de formulario Create AstraControlCenter.

El futuro

Complete el "pasos restantes" Para verificar que Astra Control Center se ha instalado correctamente, configure un controlador de entrada (opcional) e inicie sesión en la interfaz de usuario. Además, tendrá que realizar "tareas de configuración" tras completar la instalación.

Instale Astra Control Center con un back-end de almacenamiento de Cloud Volumes ONTAP

Con Astra Control Center, puede gestionar sus aplicaciones en un entorno de cloud híbrido con clústeres de Kubernetes e instancias de Cloud Volumes ONTAP autogestionados. Puede poner en marcha Astra Control Center en sus clústeres de Kubernetes en las instalaciones o en uno de los clústeres de Kubernetes autogestionados en el entorno de cloud.

Con una de estas puestas en marcha, puede realizar operaciones de gestión de datos de aplicaciones utilizando Cloud Volumes ONTAP como back-end de almacenamiento. También es posible configurar un bloque de S3 como destino de backup.

Para instalar Astra Control Center en Amazon Web Services (AWS), Google Cloud Platform (GCP) y Microsoft Azure con una back-end de almacenamiento de Cloud Volumes ONTAP, realice los siguientes pasos en función de su entorno de cloud.

- Ponga en marcha Astra Control Center en Amazon Web Services
- Ponga en marcha Astra Control Center en Google Cloud Platform
- Ponga en marcha Astra Control Center en Microsoft Azure

Puede gestionar sus aplicaciones en distribuciones con clústeres de Kubernetes autogestionados, como

OpenShift Container Platform (OCP). Sólo se validan los clústeres OCP autogestionados para la implantación de Astra Control Center.

Ponga en marcha Astra Control Center en Amazon Web Services

Puede poner en marcha Astra Control Center en un clúster de Kubernetes autogestionado alojado en un cloud público de Amazon Web Services (AWS).

Lo que necesitará para AWS

Antes de poner en marcha Astra Control Center en AWS, necesitará los siguientes elementos:

- Licencia Astra Control Center. Consulte "Requisitos de licencia de Astra Control Center".
- "Cumpla los requisitos de Astra Control Center".
- Cuenta de Cloud Central de NetApp
- Si utiliza OCP, los permisos de Red Hat OpenShift Container Platform (OCP) (en el nivel de espacio de nombres para crear pods)
- Credenciales de AWS, Access ID y Secret Key con permisos que permiten crear cubos y conectores
- Acceso e inicio de sesión del Elastic Container Registry (ECR) de la cuenta de AWS
- Se necesitan la zona alojada de AWS y la entrada de Amazon Route 53 para acceder a la interfaz de usuario de Astra Control

Requisitos de los entornos operativos para AWS

Astra Control Center requiere los siguientes entornos operativos para AWS:

- Red Hat OpenShift Container Platform 4,11 a 4,13

> Asegúrese de que el entorno operativo que elija para alojar Astra Control Center cumple los requisitos de recursos básicos que se describen en la documentación oficial del entorno.

Astra Control Center requiere los siguientes recursos además de los requisitos de recursos del entorno:

Componente	Requisito
Capacidad de almacenamiento Cloud Volumes ONTAP de back-end de NetApp	300 GB como mínimo disponible
Nodos de trabajo (requisitos de AWS EC2)	Al menos 3 nodos de trabajo en total, con 4 núcleos vCPU y 12 GB de RAM en cada uno
Equilibrador de carga	Tipo de servicio "LoadBalancer" disponible para que el tráfico de entrada se envíe a los servicios en el clúster de entorno operativo
FQDN	Método para señalar el FQDN de Astra Control Center a la dirección IP de carga equilibrada

Componente	Requisito
Astra Trident (instalado como parte de la detección de clústeres de Kubernetes en NetApp BlueXP, anteriormente Cloud Manager)	Astra Trident 23,01 o posterior instalado y configurado, y NetApp ONTAP versión 9.9.1 o posterior como back-end de almacenamiento
Registro de imágenes	NetApp proporciona un registro que puede utilizar para obtener imágenes de creación del Centro de control de Astra: `http://netappdownloads.jfrog.io/docker-astra-control-prod` Póngase en contacto con el servicio de soporte de NetApp para obtener instrucciones sobre el uso de este registro de imágenes durante el proceso de instalación del Centro de control de Astra. Si no puede acceder al registro de imágenes de NetApp, debe tener un registro privado existente, como AWS Elastic Container Registry (ECR), al que puede enviar imágenes de creación del Centro de control de Astra. Debe proporcionar la dirección URL del registro de imágenes donde cargará las imágenes. El clúster alojado de Astra Control Center y el clúster gestionado deben tener acceso al mismo registro de imágenes para poder realizar copias de seguridad y restaurar aplicaciones mediante la imagen basada en Restic.
Configuración de Astra Trident/ONTAP	Astra Control Center requiere que se cree una clase de almacenamiento y se establezca como la clase de almacenamiento predeterminada. Astra Control Center es compatible con las siguientes clases de almacenamiento ONTAP Kubernetes que se crean al importar su clúster de Kubernetes a NetApp BlueXP (anteriormente Cloud Manager). Los proporciona Astra Trident: • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

Estos requisitos suponen que Astra Control Center es la única aplicación que se ejecuta en el entorno operativo. Si el entorno ejecuta aplicaciones adicionales, ajuste estos requisitos mínimos según corresponda.

El token del registro de AWS caduca en 12 horas, después del cual deberá renovar el secreto del registro de imagen Docker.

Información general sobre la implementación para AWS

He aquí una descripción general del proceso de instalación de Astra Control Center para AWS con Cloud Volumes ONTAP como back-end de almacenamiento.

Cada uno de estos pasos se explica más detalladamente a continuación.

1. Compruebe que dispone de suficientes permisos IAM.
2. Instale un clúster RedHat OpenShift en AWS.
3. Configure AWS.
4. Configuración de BlueXP de NetApp para AWS.
5. Instale Astra Control Center para AWS.

Compruebe que dispone de suficientes permisos IAM

Asegúrese de tener suficientes roles y permisos de IAM que le permiten instalar un clúster RedHat OpenShift y un conector BlueXP de NetApp (anteriormente Cloud Manager).

Consulte "Credenciales iniciales de AWS".

Instale un clúster RedHat OpenShift en AWS

Instale un clúster RedHat OpenShift Container Platform en AWS.

Para obtener instrucciones de instalación, consulte "Instalación de un clúster en AWS en OpenShift Container Platform".

Configure AWS

A continuación, configure AWS para crear una red virtual, configurar EC2 instancias de computación y crear un bucket de AWS S3. Si no puede acceder a Registro de imágenes del Centro de control de Astra de NetApp, También tendrá que crear un registro de contenedores elásticos (ECR) para alojar las imágenes de Astra Control Center y enviar las imágenes a este registro.

Siga la documentación de AWS para completar los pasos siguientes. Consulte "Documentación de instalación de AWS".

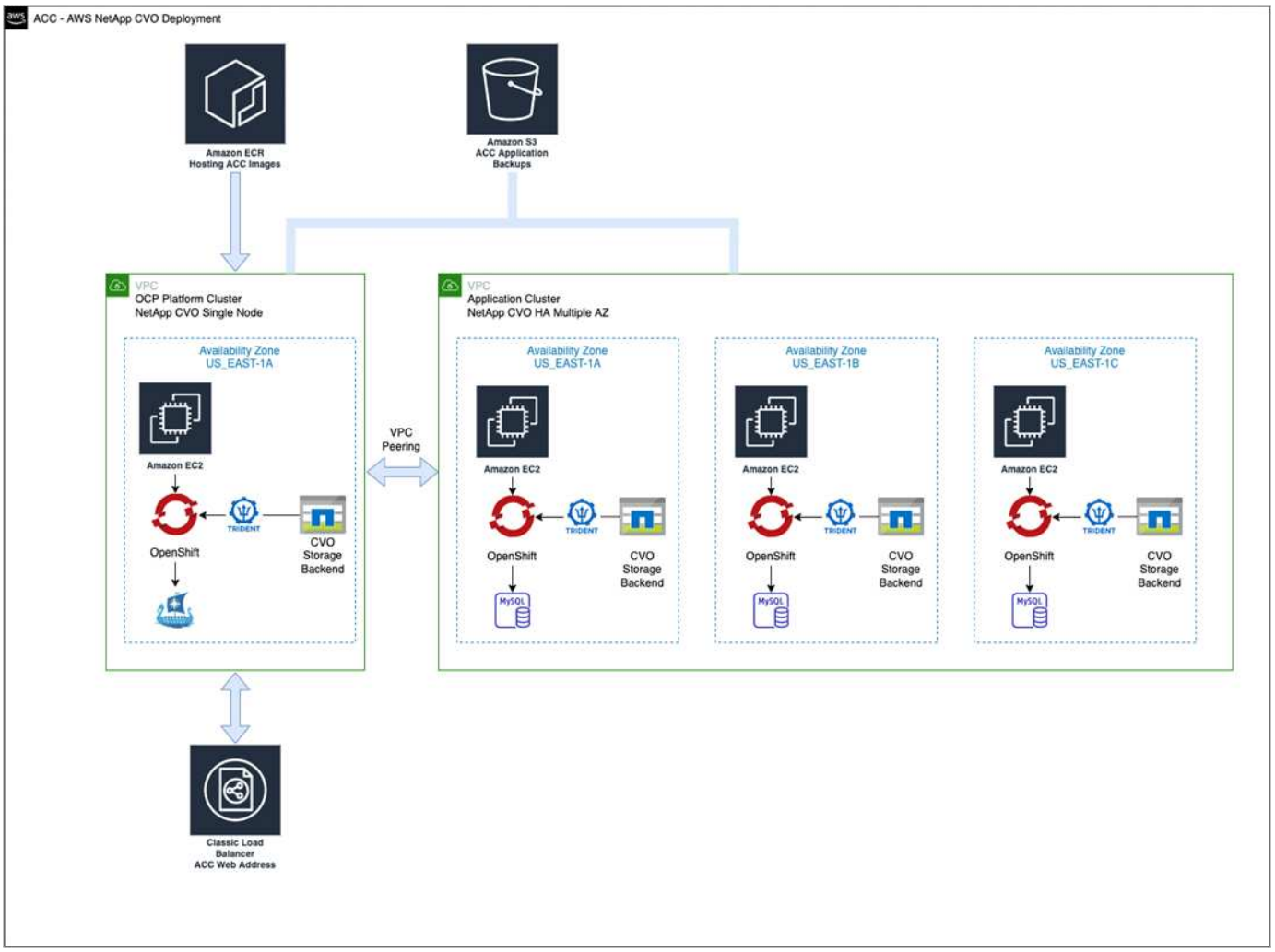
1. Cree una red virtual AWS.
2. Revise las instancias de computación EC2. Puede ser un servidor con configuración básica o máquinas virtuales en AWS.
3. Si el tipo de instancia no coincide con los requisitos mínimos de recursos de Astra para los nodos maestros y trabajadores, cambie el tipo de instancia en AWS para cumplir los requisitos de Astra. Consulte "Requisitos del Centro de Control de Astra".
4. Cree al menos un bloque de AWS S3 para almacenar los backups.
5. (Opcional) Si no puede acceder al Registro de imágenes de NetApp, haga lo siguiente:
    a. Cree un AWS Elastic Container Registry (ECR) para alojar todas las imágenes de Astra Control Center.

Si no crea la ECR, Astra Control Center no puede acceder a los datos de supervisión de un clúster que contenga Cloud Volumes ONTAP con un back-end de AWS. El problema se produce cuando el clúster que intenta detectar y gestionar mediante Astra Control Center no tiene acceso a AWS ECR.

b. Envía las imágenes del Centro de control de Astra al registro definido.

El token del registro de contenedor elástico de AWS (ECR) caduca al cabo de 12 horas y provoca errores en las operaciones de clonado de varios clústeres. Este problema ocurre cuando se gestiona un back-end de almacenamiento desde Cloud Volumes ONTAP configurado para AWS. Para corregir este problema, vuelva a autenticarse con la ECR y genere un nuevo secreto para que las operaciones de clonación se reanuden correctamente.

A continuación mostramos un ejemplo de una puesta en marcha de AWS:



Configuración de BlueXP de NetApp para AWS

Con NetApp BlueXP (anteriormente Cloud Manager), cree un espacio de trabajo, añada un conector a AWS, cree un entorno de trabajo e importe el clúster.

Siga la documentación de BlueXP para completar los siguientes pasos. Consulte lo siguiente:

- "Introducción a Cloud Volumes ONTAP en AWS".
- "Cree un conector en AWS mediante BlueXP"

Pasos

1. Agregue sus credenciales a BlueXP.
2. Crear un área de trabajo.
3. Agregue un conector para AWS. Elija AWS como proveedor.
4. Cree un entorno de trabajo para su entorno de cloud.
   a. Ubicación: «Amazon Web Services (AWS)»
   b. Tipo: "Cloud Volumes ONTAP ha"
5. Importe el clúster OpenShift. El clúster se conectará al entorno de trabajo que acaba de crear.
   a. Consulte los detalles del clúster de NetApp seleccionando **K8s** > **Lista de clústeres** > **Detalles del clúster**.
   b. En la esquina superior derecha, observa la versión de Astra Trident.
   c. Observe las clases de almacenamiento del clúster Cloud Volumes ONTAP que muestran NetApp como el aprovisionador.

      Esto importa su clúster de Red Hat OpenShift y le asigna una clase de almacenamiento predeterminada. Seleccione la clase de almacenamiento.
      Astra Trident se instala automáticamente como parte del proceso de importación y detección.
6. Obsérvese todos los volúmenes y volúmenes persistentes en esta puesta en marcha de Cloud Volumes ONTAP.

> Cloud Volumes ONTAP puede funcionar como un nodo único o en alta disponibilidad. Si está habilitada, anote el estado de alta disponibilidad y el estado de implementación del nodo que se ejecutan en AWS.

Instale Astra Control Center para AWS

Siga la norma "Instrucciones de instalación de Astra Control Center".

> AWS utiliza el tipo de bloque Generic S3.

Ponga en marcha Astra Control Center en Google Cloud Platform

Puede poner en marcha Astra Control Center en un clúster de Kubernetes autogestionado alojado en un cloud público de Google Cloud Platform (GCP).

Qué necesitará para GCP

Antes de implementar Astra Control Center en GCP, necesitará los siguientes elementos:

- Licencia Astra Control Center. Consulte "Requisitos de licencia de Astra Control Center".
- "Cumpla los requisitos de Astra Control Center".
- Cuenta de Cloud Central de NetApp
- Si utiliza OCP, Red Hat OpenShift Container Platform (OCP) 4,11 a 4,13

- Si utiliza OCP, los permisos de Red Hat OpenShift Container Platform (OCP) (en el nivel de espacio de nombres para crear pods)
- Cuenta de servicio de GCP con permisos que le permiten crear cubos y conectores

Requisitos del entorno operativo para GCP

Asegúrese de que el entorno operativo que elija para alojar Astra Control Center cumple los requisitos de recursos básicos que se describen en la documentación oficial del entorno.

Astra Control Center requiere los siguientes recursos además de los requisitos de recursos del entorno:

Componente	Requisito
Capacidad de almacenamiento Cloud Volumes ONTAP de back-end de NetApp	300 GB como mínimo disponible
Nodos de trabajo (requisitos de computación de GCP)	Al menos 3 nodos de trabajo en total, con 4 núcleos vCPU y 12 GB de RAM en cada uno
Equilibrador de carga	Tipo de servicio "LoadBalancer" disponible para que el tráfico de entrada se envíe a los servicios en el clúster de entorno operativo
FQDN (ZONA DNS DE GCP)	Método para señalar el FQDN de Astra Control Center a la dirección IP de carga equilibrada
Astra Trident (instalado como parte de la detección de clústeres de Kubernetes en NetApp BlueXP, anteriormente Cloud Manager)	Astra Trident 23,01 o posterior instalado y configurado, y NetApp ONTAP versión 9.9.1 o posterior como back-end de almacenamiento
Registro de imágenes	NetApp proporciona un registro que puede utilizar para obtener imágenes de creación del Centro de control de Astra: `http://netappdownloads.jfrog.io/docker-astra-control-prod` Póngase en contacto con el servicio de soporte de NetApp para obtener instrucciones sobre el uso de este registro de imágenes durante el proceso de instalación del Centro de control de Astra. Si no puede acceder al registro de imágenes de NetApp, debe tener un registro privado existente, como el Registro de contenedores de Google, en el que puede enviar imágenes de creación del Centro de control de Astra. Debe proporcionar la dirección URL del registro de imágenes donde cargará las imágenes. Es necesario habilitar el acceso anónimo para extraer imágenes RTIC para realizar copias de seguridad.

Componente	Requisito
Configuración de Astra Trident/ONTAP	Astra Control Center requiere que se cree una clase de almacenamiento y se establezca como la clase de almacenamiento predeterminada. Astra Control Center es compatible con las siguientes clases de almacenamiento ONTAP Kubernetes que se crean al importar su clúster Kubernetes a BlueXP de NetApp. Los proporciona Astra Trident: • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

Estos requisitos suponen que Astra Control Center es la única aplicación que se ejecuta en el entorno operativo. Si el entorno ejecuta aplicaciones adicionales, ajuste estos requisitos mínimos según corresponda.

Información general de puesta en marcha para GCP

A continuación se ofrece una descripción general del proceso de instalación de Astra Control Center en un clúster OCP autogestionado en GCP con Cloud Volumes ONTAP como back-end de almacenamiento.

Cada uno de estos pasos se explica más detalladamente a continuación.

1. Instalar un clúster RedHat OpenShift en GCP.
2. Cree un proyecto de GCP y una nube privada virtual.
3. Compruebe que dispone de suficientes permisos IAM.
4. Configurar GCP.
5. Configuración de NetApp BlueXP para GCP.
6. Instala Astra Control Center para GCP.

Instalar un clúster RedHat OpenShift en GCP

El primer paso es instalar un clúster RedHat OpenShift en GCP.

Para obtener instrucciones de instalación, consulte lo siguiente:

- "Instalar un clúster OpenShift en GCP"
- "Creación de una cuenta de servicio de GCP"

Cree un proyecto de GCP y una nube privada virtual

Cree al menos un proyecto de GCP y una nube privada virtual (VPC).

OpenShift podría crear sus propios grupos de recursos. Además de ellas, debe definir también un VPC de GCP. Consulte la documentación de OpenShift.

Es posible que desee crear un grupo de recursos de clúster de plataforma y un grupo de recursos de clúster de aplicación OpenShift de destino.

Compruebe que dispone de suficientes permisos IAM

Asegúrese de tener suficientes roles y permisos de IAM que le permiten instalar un clúster RedHat OpenShift y un conector BlueXP de NetApp (anteriormente Cloud Manager).

Consulte "Credenciales y permisos iniciales de GCP".

Configurar GCP

A continuación, configure GCP para crear una VPC, configurar instancias de computación y crear un almacenamiento de objetos de Google Cloud. Si no puede acceder a Registro de imágenes del Centro de control de Astra de NetApp, También tendrá que crear un Registro de contenedores de Google para alojar las imágenes del Centro de control de Astra y enviar las imágenes a este registro.

Siga la documentación de GCP para completar los siguientes pasos. Consulte instalación del clúster OpenShift en GCP.

1. Cree un proyecto de GCP y VPC en el GCP que planea utilizar para el clúster de OCP con el back-end de CVO.
2. Revise las instancias de computación. Puede tratarse de un servidor de configuración básica o máquinas virtuales en GCP.
3. Si el tipo de instancia no coincide con los requisitos mínimos de recursos de Astra para los nodos maestro y trabajador, cambie el tipo de instancia de GCP para que cumpla los requisitos de Astra. Consulte "Requisitos del Centro de Control de Astra".
4. Cree al menos un bloque de almacenamiento en cloud de GCP para almacenar sus backups.
5. Crear un secreto, que es necesario para el acceso a bloques.
6. (Opcional) Si no puede acceder al Registro de imágenes de NetApp, haga lo siguiente:

    a. Crea un registro de contenedores de Google para alojar las imágenes del Centro de control de Astra.
    b. Configure el acceso al registro de contenedores de Google para inserción/extracción de Docker para todas las imágenes de Astra Control Center.

    Ejemplo: Las imágenes del Centro de control de Astra se pueden enviar a este registro introduciendo el siguiente script:

    ```
    gcloud auth activate-service-account <service account email address>
    --key-file=<GCP Service Account JSON file>
    ```

    Este script requiere un archivo de manifiesto de Astra Control Center y su ubicación del Registro de imágenes de Google. Ejemplo:

```
manifestfile=acc.manifest.bundle.yaml
GCP_CR_REGISTRY=<target GCP image registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image
    docker push $GCP_CR_REGISTRY/$image
done < acc.manifest.bundle.yaml
```

7. Configure zonas DNS.

Configuración de NetApp BlueXP para GCP

Utilizando NetApp BlueXP (anteriormente Cloud Manager), crear un espacio de trabajo, añadir un conector a GCP, crear un entorno de trabajo e importar el clúster.

Siga la documentación de BlueXP para completar los siguientes pasos. Consulte "Primeros pasos con Cloud Volumes ONTAP en GCP".

Antes de empezar

- Acceso a la cuenta de servicio de GCP con los permisos y roles de IAM necesarios

Pasos

1. Agregue sus credenciales a BlueXP. Consulte "Adición de cuentas de GCP".
2. Añade un conector para GCP.
   a. Elija "GCP" como el proveedor.
   b. Introduzca las credenciales de GCP. Consulte "Creación de un conector en GCP desde BlueXP".
   c. Asegúrese de que el conector está en marcha y cambie a dicho conector.
3. Cree un entorno de trabajo para su entorno de cloud.
   a. Ubicación: «GCP»
   b. Tipo: "Cloud Volumes ONTAP ha"
4. Importe el clúster OpenShift. El clúster se conectará al entorno de trabajo que acaba de crear.
   a. Consulte los detalles del clúster de NetApp seleccionando **K8s** > **Lista de clústeres** > **Detalles del clúster**.
   b. En la esquina superior derecha, tenga en cuenta la versión de Trident.
   c. Tenga en cuenta las clases de almacenamiento del clúster de Cloud Volumes ONTAP que muestran "NetApp" como el aprovisionador.

      Esto importa su clúster de Red Hat OpenShift y le asigna una clase de almacenamiento predeterminada. Seleccione la clase de almacenamiento.
      Astra Trident se instala automáticamente como parte del proceso de importación y detección.

5. Obsérvese todos los volúmenes y volúmenes persistentes en esta puesta en marcha de Cloud Volumes ONTAP.

> Cloud Volumes ONTAP puede funcionar como un nodo único o en alta disponibilidad. Si está habilitada, anote el estado de alta disponibilidad y el estado de puesta en marcha del nodo que se ejecutan en GCP.

Instala Astra Control Center para GCP

Siga la norma "Instrucciones de instalación de Astra Control Center".

> GCP utiliza el tipo de bucket Generic S3.

1. Genere el secreto Docker para obtener imágenes de la instalación de Astra Control Center:

```
kubectl create secret docker-registry <secret name> --docker
-server=<Registry location> --docker-username=_json_key --docker
-password="$(cat <GCP Service Account JSON file>)" --namespace=pcloud
```

Ponga en marcha Astra Control Center en Microsoft Azure

Puede poner en marcha Astra Control Center en un clúster de Kubernetes autogestionado que se aloja en un cloud público de Microsoft Azure.

Lo que necesitará para Azure

Antes de poner en marcha Astra Control Center en Azure, necesitará los siguientes elementos:

- Licencia Astra Control Center. Consulte "Requisitos de licencia de Astra Control Center".
- "Cumpla los requisitos de Astra Control Center".
- Cuenta de Cloud Central de NetApp
- Si utiliza OCP, Red Hat OpenShift Container Platform (OCP) 4,11 a 4,13
- Si utiliza OCP, los permisos de Red Hat OpenShift Container Platform (OCP) (en el nivel de espacio de nombres para crear pods)
- Credenciales de Azure con permisos que le permiten crear cubos y conectores

Requisitos del entorno operativo para Azure

Asegúrese de que el entorno operativo que elija para alojar Astra Control Center cumple los requisitos de recursos básicos que se describen en la documentación oficial del entorno.

Astra Control Center requiere los siguientes recursos además de los requisitos de recursos del entorno:

Consulte "Requisitos del entorno operativo del Centro de control de Astra".

Componente	Requisito
Capacidad de almacenamiento Cloud Volumes ONTAP de back-end de NetApp	300 GB como mínimo disponible
Nodos de trabajo (requisitos de computación de Azure)	Al menos 3 nodos de trabajo en total, con 4 núcleos vCPU y 12 GB de RAM en cada uno
Equilibrador de carga	Tipo de servicio "LoadBalancer" disponible para que el tráfico de entrada se envíe a los servicios en el clúster de entorno operativo
FQDN (zona DNS de Azure)	Método para señalar el FQDN de Astra Control Center a la dirección IP de carga equilibrada
Astra Trident (instalado como parte de la detección de clústeres de Kubernetes en NetApp BlueXP)	Astra Trident 23,01 o posterior instalado y configurado, y NetApp ONTAP versión 9.9.1 o posterior se utilizarán como back-end de almacenamiento
Registro de imágenes	NetApp proporciona un registro que puede utilizar para obtener imágenes de creación del Centro de control de Astra: `http://netappdownloads.jfrog.io/docker-astra-control-prod` Póngase en contacto con el servicio de soporte de NetApp para obtener instrucciones sobre el uso de este registro de imágenes durante el proceso de instalación del Centro de control de Astra. Si no puede acceder al registro de imágenes de NetApp, debe tener un registro privado existente, como Azure Container Registry (ACR), en el que puede insertar imágenes de creación del Centro de control de Astra. Debe proporcionar la dirección URL del registro de imágenes donde cargará las imágenes. Es necesario habilitar el acceso anónimo para extraer imágenes RTIC para realizar copias de seguridad.

Componente	Requisito
Configuración de Astra Trident/ONTAP	Astra Control Center requiere que se cree una clase de almacenamiento y se establezca como la clase de almacenamiento predeterminada. Astra Control Center es compatible con las siguientes clases de almacenamiento ONTAP Kubernetes que se crean al importar su clúster Kubernetes a BlueXP de NetApp. Los proporciona Astra Trident: • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

Estos requisitos suponen que Astra Control Center es la única aplicación que se ejecuta en el entorno operativo. Si el entorno ejecuta aplicaciones adicionales, ajuste estos requisitos mínimos según corresponda.

Información general sobre la implementación para Azure

A continuación se ofrece una descripción general del proceso de instalación de Astra Control Center para Azure.

Cada uno de estos pasos se explica más detalladamente a continuación.

1. Instale un clúster RedHat OpenShift en Azure.
2. Cree grupos de recursos de Azure.
3. Compruebe que dispone de suficientes permisos IAM.
4. Configure Azure.
5. Configuración de NetApp BlueXP (anteriormente Cloud Manager) para Azure.
6. Instalar y configurar Astra Control Center para Azure.

Instale un clúster RedHat OpenShift en Azure

El primer paso es instalar un clúster RedHat OpenShift en Azure.

Para obtener instrucciones de instalación, consulte lo siguiente:

- "Instalando el clúster de OpenShift en Azure".
- "Instalar una cuenta de Azure".

Cree grupos de recursos de Azure

Cree al menos un grupo de recursos de Azure.

OpenShift podría crear sus propios grupos de recursos. Además de estos, también debe definir los grupos de recursos de Azure. Consulte la documentación de OpenShift.

Es posible que desee crear un grupo de recursos de clúster de plataforma y un grupo de recursos de clúster de aplicación OpenShift de destino.

Compruebe que dispone de suficientes permisos IAM

Asegúrese de tener suficientes roles y permisos IAM para poder instalar un clúster RedHat OpenShift y un conector BlueXP de NetApp.

Consulte "Credenciales y permisos de Azure".

Configure Azure

A continuación, configure Azure para crear una red virtual, configurar instancias de computación y crear un contenedor de Azure Blob. Si no puede acceder a Registro de imágenes del Centro de control de Astra de NetApp, También tendrá que crear un Azure Container Registry (ACR) para alojar las imágenes de Astra Control Center y enviar las imágenes a este registro.

Siga la documentación de Azure para completar los siguientes pasos. Consulte "Instalando el clúster de OpenShift en Azure".

1. Cree una red virtual de Azure.
2. Revise las instancias de computación. Puede ser un servidor con configuración básica o máquinas virtuales en Azure.
3. Si el tipo de instancia no coincide con los requisitos mínimos de recursos de Astra para los nodos maestros y trabajadores, cambie el tipo de instancia en Azure para cumplir los requisitos de Astra. Consulte "Requisitos del Centro de Control de Astra".
4. Cree al menos un contenedor de Azure Blob para almacenar los backups.
5. Cree una cuenta de almacenamiento. Necesitará una cuenta de almacenamiento para crear un contenedor que se utilizará como bloque en Astra Control Center.
6. Crear un secreto, que es necesario para el acceso a bloques.
7. (Opcional) Si no puede acceder al Registro de imágenes de NetApp, haga lo siguiente:
    a. Cree un registro de contenedores de Azure (ACR) para alojar las imágenes del Centro de control de Astra.
    b. Configura el acceso de ACR para la inserción/extracción de Docker para todas las imágenes del Centro de control de Astra.
    c. Envíe las imágenes del Centro de control de Astra a este registro mediante el siguiente script:

```
az acr login -n <AZ ACR URL/Location>
This script requires the Astra Control Center manifest file and your
Azure ACR location.
```

Ejemplo:

```
manifestfile=acc.manifest.bundle.yaml
AZ_ACR_REGISTRY=<target Azure ACR image registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image
    docker push $AZ_ACR_REGISTRY/$image
done < acc.manifest.bundle.yaml
```

8. Configure zonas DNS.

Configuración de NetApp BlueXP (anteriormente Cloud Manager) para Azure

Con BlueXP (anteriormente Cloud Manager), cree un espacio de trabajo, añada un conector a Azure, cree un entorno de trabajo e importe el clúster.

Siga la documentación de BlueXP para completar los siguientes pasos. Consulte "Introducción a BlueXP en Azure".

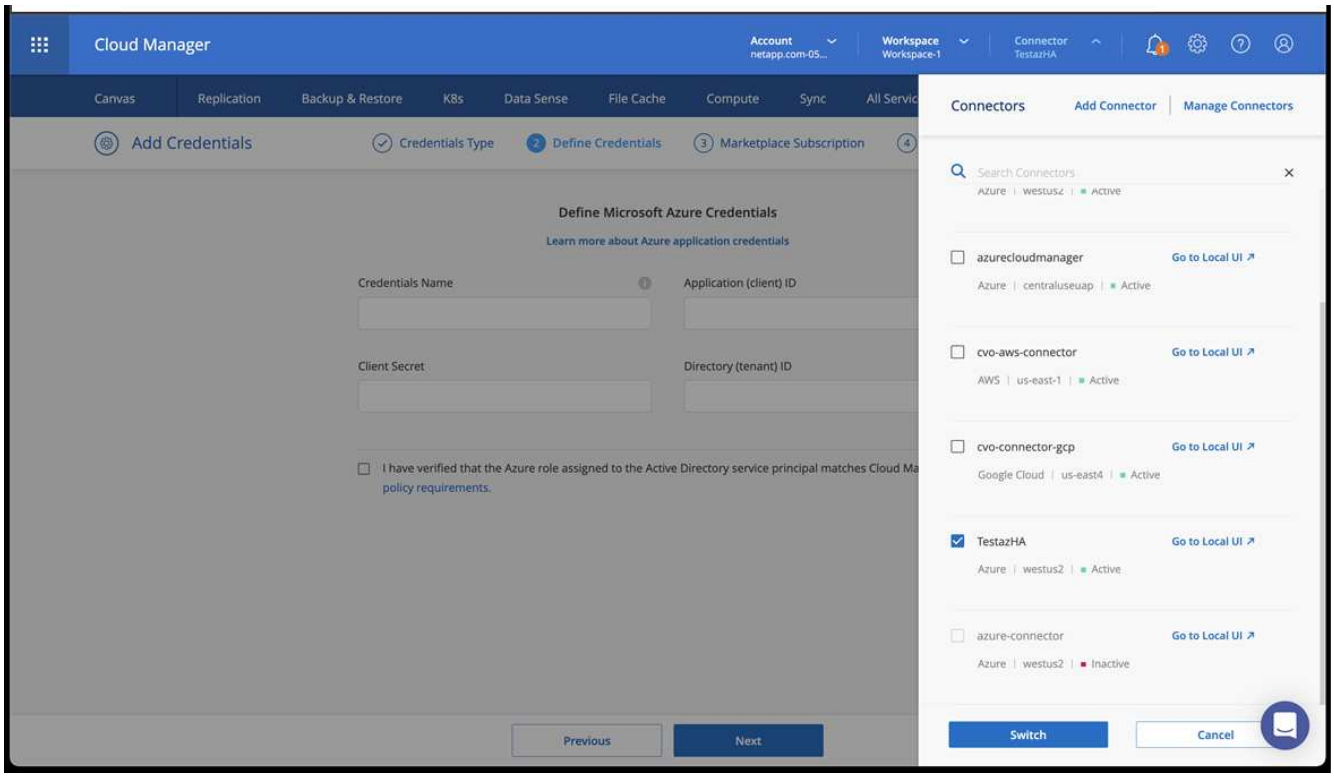
Antes de empezar

Acceso a la cuenta de Azure con los permisos y roles de IAM necesarios

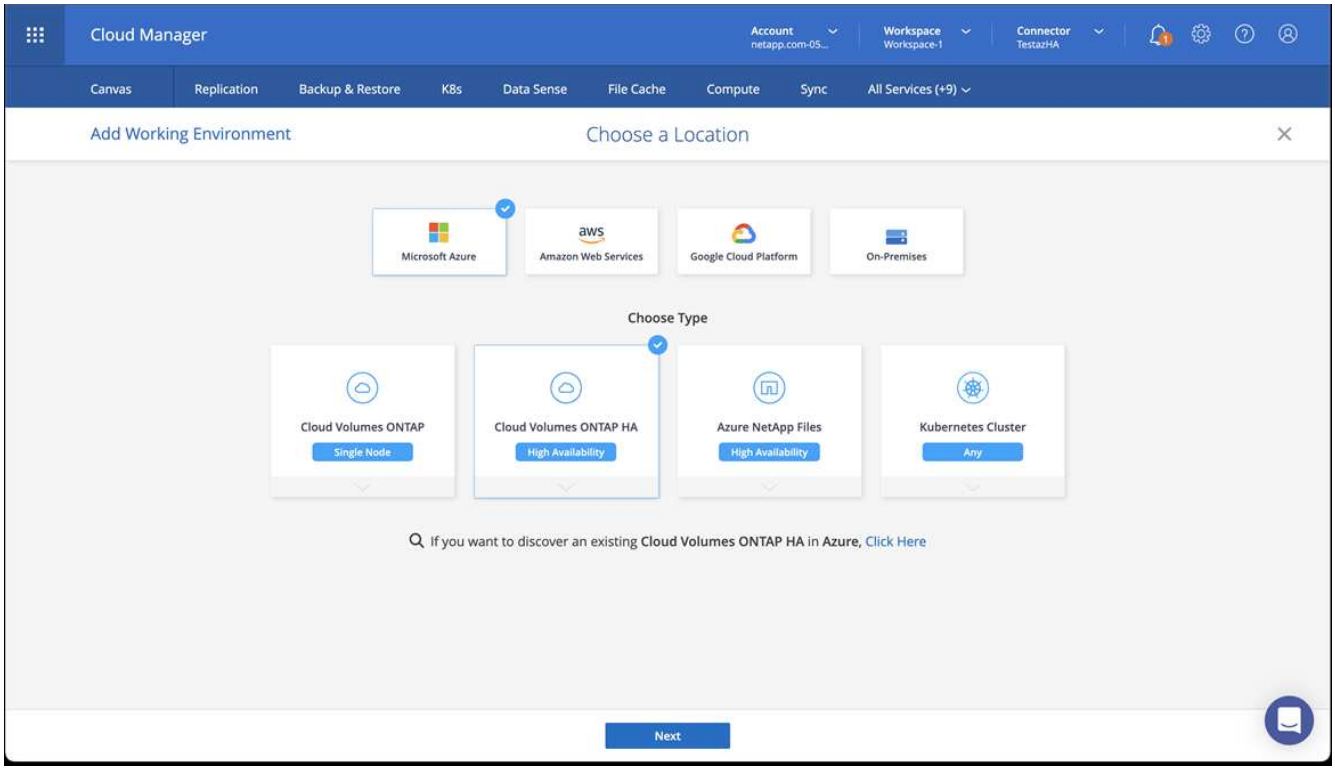
Pasos

1. Agregue sus credenciales a BlueXP.
2. Agregue un conector para Azure. Consulte "Políticas de BlueXP".
   a. Elija **Azure** como proveedor.
   b. Introduzca las credenciales de Azure, incluidos el ID de aplicación, el secreto de cliente y el ID del directorio (inquilino).

      Consulte "Creación de un conector en Azure desde BlueXPr".
3. Asegúrese de que el conector está en marcha y cambie a dicho conector.

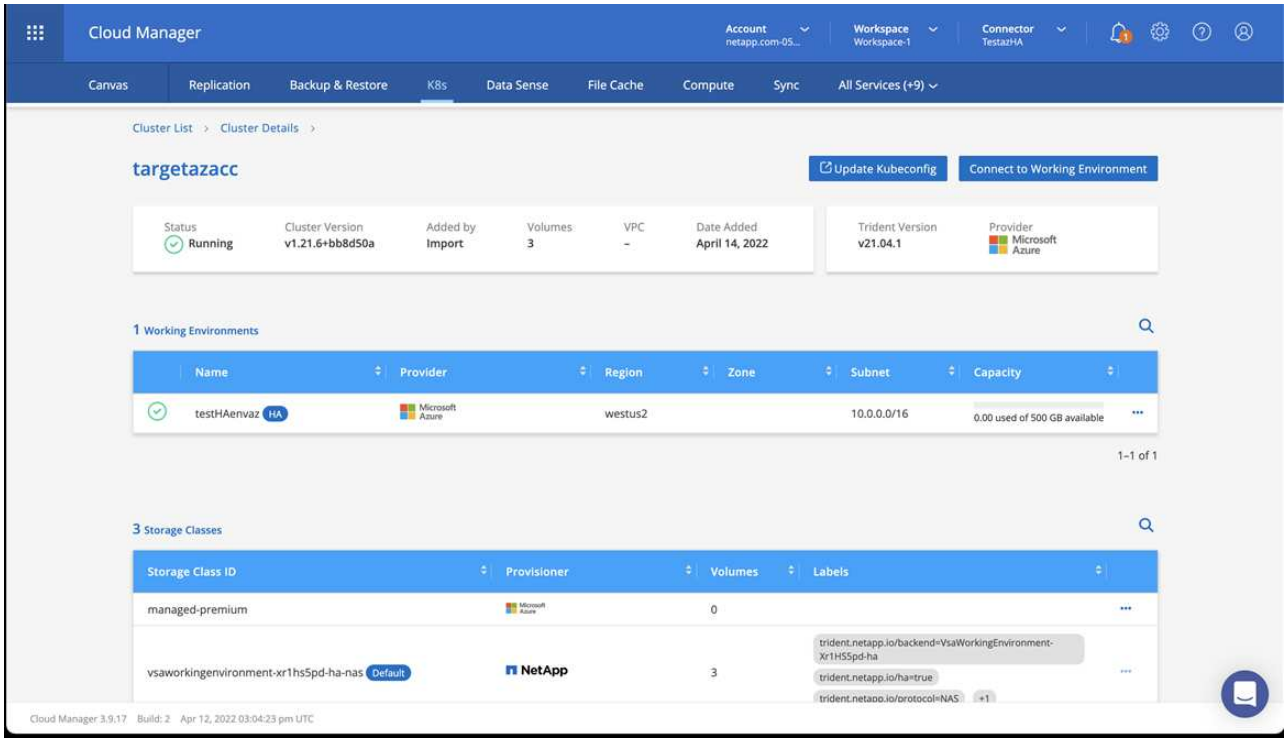


4. Cree un entorno de trabajo para su entorno de cloud.
   a. Ubicación: "Microsoft Azure".
   b. Tipo: "Cloud Volumes ONTAP ha".



5. Importe el clúster OpenShift. El clúster se conectará al entorno de trabajo que acaba de crear.
   a. Consulte los detalles del clúster de NetApp seleccionando **K8s** > **Lista de clústeres** > **Detalles del**

clúster.



b. En la esquina superior derecha, observa la versión de Astra Trident.

c. Observe las clases de almacenamiento del clúster Cloud Volumes ONTAP que muestran NetApp como el aprovisionador.

Esto importa su clúster de Red Hat OpenShift y asigna una clase de almacenamiento predeterminada. Seleccione la clase de almacenamiento.
Astra Trident se instala automáticamente como parte del proceso de importación y detección.

6. Obsérvese todos los volúmenes y volúmenes persistentes en esta puesta en marcha de Cloud Volumes ONTAP.
7. Cloud Volumes ONTAP puede funcionar como un nodo único o en alta disponibilidad. Si ha está habilitada, anote el estado de alta disponibilidad y el estado de puesta en marcha del nodo que se ejecutan en Azure.

Instalar y configurar Astra Control Center para Azure

Instale Astra Control Center con el estándar "instrucciones de instalación".

Con Astra Control Center, añada un bucket de Azure. Consulte "Configure Astra Control Center y añada cucharones".

Configurar Astra Control Center después de la instalación

En función de su entorno, es posible que se necesite una configuración adicional después de instalar Astra Control Center.

Quite las limitaciones de recursos

Algunos entornos utilizan los objetos ResourceQuotas y LimitRanges para evitar que los recursos de un espacio de nombres consuman toda la CPU y memoria disponibles en el clúster. Astra Control Center no

establece límites máximos, por lo que no se ajusta a esos recursos. Si su entorno se configura de esta forma, debe eliminar esos recursos de los espacios de nombres en los que planea instalar Astra Control Center.

Puede utilizar los siguientes pasos para recuperar y eliminar estas cuotas y límites. En estos ejemplos, el resultado del comando se muestra inmediatamente después del comando.

Pasos

1. Obtenga las cuotas de recursos en la `netapp-acc` espacio de nombres (o con nombre personalizado):

```
kubectl get quota -n [netapp-acc or custom namespace]
```

Respuesta:

```
NAME          AGE   REQUEST                                              LIMIT
pods-high     16s   requests.cpu: 0/20, requests.memory: 0/100Gi
limits.cpu: 0/200, limits.memory: 0/1000Gi
pods-low      15s   requests.cpu: 0/1, requests.memory: 0/1Gi
limits.cpu: 0/2, limits.memory: 0/2Gi
pods-medium   16s   requests.cpu: 0/10, requests.memory: 0/20Gi
limits.cpu: 0/20, limits.memory: 0/200Gi
```

2. Elimine todas las cuotas de recursos por nombre:

```
kubectl delete resourcequota pods-high -n [netapp-acc or custom
namespace]
```

```
kubectl delete resourcequota pods-low -n [netapp-acc or custom
namespace]
```

```
kubectl delete resourcequota pods-medium -n [netapp-acc or custom
namespace]
```

3. Obtenga los rangos de límites en la `netapp-acc` espacio de nombres (o con nombre personalizado):

```
kubectl get limits -n [netapp-acc or custom namespace]
```

Respuesta:

```
NAME              CREATED AT
cpu-limit-range   2022-06-27T19:01:23Z
```

4. Eliminar los rangos de límites por nombre:

```
kubectl delete limitrange cpu-limit-range -n [netapp-acc or custom
namespace]
```

Agregue un certificado TLS personalizado

Astra Control Center utiliza un certificado TLS autofirmado de forma predeterminada para el tráfico del controlador de entrada (solo en determinadas configuraciones) y la autenticación de la interfaz de usuario web con exploradores web. Puede quitar el certificado TLS autofirmado existente y reemplazarlo con un certificado TLS firmado por una entidad de certificación (CA).

El certificado autofirmado predeterminado se utiliza para dos tipos de conexiones:

- Conexiones HTTPS a la interfaz de usuario web de Astra Control Center
- Tráfico del controlador de entrada (sólo si el `ingressType: "AccTraefik"` la propiedad se estableció en `astra_control_center.yaml` Archivo durante la instalación de Astra Control Center)

Al reemplazar el certificado TLS predeterminado, se reemplaza el certificado utilizado para la autenticación de estas conexiones.

Antes de empezar

- Clúster Kubernetes con Astra Control Center instalado
- Acceso administrativo a un shell de comandos en el clúster para ejecutar `kubectl` comandos
- Archivos de claves privadas y certificados de la CA

Quite el certificado autofirmado

Quite el certificado TLS autofirmado existente.

1. Con SSH, inicie sesión en el clúster Kubernetes que aloja Astra Control Center como usuario administrativo.
2. Busque el secreto TLS asociado con el certificado actual mediante el comando siguiente, reemplazo `<ACC-deployment-namespace>` Con el espacio de nombres de puesta en marcha de Astra Control Center:

```
kubectl get certificate -n <ACC-deployment-namespace>
```

3. Elimine el secreto y certificado instalados actualmente con los comandos siguientes:

```
kubectl delete cert cert-manager-certificates -n <ACC-deployment-
namespace>
```

```
kubectl delete secret secure-testing-cert -n <ACC-deployment-namespace>
```

Añada un nuevo certificado mediante la línea de comandos

Añada un nuevo certificado TLS firmado por una CA.

1. Utilice el siguiente comando para crear el nuevo secreto TLS con la clave privada y los archivos de certificado de la CA, reemplazando los argumentos entre paréntesis <> con la información adecuada:

```
kubectl create secret tls <secret-name> --key <private-key-filename>
--cert <certificate-filename> -n <ACC-deployment-namespace>
```

2. Utilice el siguiente comando y el ejemplo para editar el archivo de definición de recursos personalizados (CRD) del clúster y cambiar el `spec.selfSigned` valor a. `spec.ca.secretName` Para hacer referencia al secreto TLS que ha creado anteriormente:

```
kubectl edit clusterissuers.cert-manager.io/cert-manager-certificates -n
<ACC-deployment-namespace>
```

CRD:

```
#spec:
#  selfSigned: {}

spec:
  ca:
    secretName: <secret-name>
```

3. Utilice el siguiente comando y el resultado de ejemplo para validar que los cambios son correctos y que el clúster está listo para validar certificados, sustituir `<ACC-deployment-namespace>` Con el espacio de nombres de puesta en marcha de Astra Control Center:

```
kubectl describe clusterissuers.cert-manager.io/cert-manager-
certificates -n <ACC-deployment-namespace>
```

Respuesta:

```
Status:
  Conditions:
    Last Transition Time:  2021-07-01T23:50:27Z
    Message:               Signing CA verified
    Reason:                KeyPairVerified
    Status:                True
    Type:                  Ready
Events:                    <none>
```

4. Cree el `certificate.yaml` archivo utilizando el ejemplo siguiente, reemplazando los valores de marcador de posición entre corchetes <> con la información apropiada:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  <strong>name: <certificate-name></strong>
  namespace: <ACC-deployment-namespace>
spec:
  <strong>secretName: <certificate-secret-name></strong>
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  dnsNames:
  <strong>- <astra.dnsname.example.com></strong> #Replace with the
correct Astra Control Center DNS address
  issuerRef:
    kind: ClusterIssuer
    name: cert-manager-certificates
```

5. Cree el certificado con el comando siguiente:

```
kubectl apply -f certificate.yaml
```

6. Con el siguiente comando y el resultado de ejemplo, valide que el certificado se ha creado correctamente y con los argumentos especificados durante la creación (como nombre, duración, plazo de renovación y nombres DNS).

```
kubectl describe certificate -n <ACC-deployment-namespace>
```

Respuesta:

```
Spec:
  Dns Names:
    astra.example.com
  Duration:  125h0m0s
  Issuer Ref:
    Kind:        ClusterIssuer
    Name:        cert-manager-certificates
  Renew Before:  61h0m0s
  Secret Name:   <certificate-secret-name>
Status:
  Conditions:
    Last Transition Time:  2021-07-02T00:45:41Z
    Message:               Certificate is up to date and has not expired
    Reason:                Ready
    Status:                True
    Type:                  Ready
  Not After:               2021-07-07T05:45:41Z
  Not Before:              2021-07-02T00:45:41Z
  Renewal Time:            2021-07-04T16:45:41Z
  Revision:                1
Events:                    <none>
```

7. Edite el almacén de CRD de TLS para que apunte al nuevo nombre de secreto de certificado mediante el siguiente comando y por ejemplo, sustituyendo los valores de marcador de posición entre paréntesis <> por la información adecuada

```
kubectl edit tlsstores.traefik.io -n <ACC-deployment-namespace>
```

CRD:

```
...
spec:
  defaultCertificate:
    secretName: <certificate-secret-name>
```

8. Edite la opción Ingress CRD TLS para que apunte al nuevo secreto de certificado utilizando el siguiente comando y ejemplo, reemplazando los valores de marcador de posición entre paréntesis <> con la información adecuada:

```
kubectl edit ingressroutes.traefik.io -n <ACC-deployment-namespace>
```

CRD:

```
...
 tls:
    secretName: <certificate-secret-name>
```

9. Con un explorador web, vaya a la dirección IP de implementación de Astra Control Center.
10. Compruebe que los detalles del certificado coinciden con los detalles del certificado que ha instalado.
11. Exporte el certificado e importe el resultado en el administrador de certificados en su navegador web.