Instalar Astra Control Center

Astra Control Center

NetApp
April 25, 2024

Tabla de contenidos

Instale Astra Control Center mediante el proceso estándar

Para instalar Astra Control Center, descargue las imágenes de instalación y siga estos pasos. Puede utilizar este procedimiento para instalar Astra Control Center en entornos conectados a Internet o con conexión por aire.

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.
- **(Solo controlador SAN de ONTAP) Habilitar acceso múltiple**: Si está utilizando un controlador SAN de ONTAP, asegúrese de que la opción multivía esté habilitada en todos sus clústeres de Kubernetes.

También debe tener en cuenta lo siguiente:

- **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.

  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>
```

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

  > La integración de Astra Control Center con una malla de servicios solo puede llevarse a cabo durante Astra Control Center "instalación" y no independiente de este proceso. No se admite el cambio de un entorno mallado a otro sin mallado.

  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
```

Pasos

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

- Descargue y extraiga Astra Control Center
- Complete los pasos adicionales si utiliza un 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

Descargue las imágenes del Centro de control de Astra de una de las siguientes ubicaciones:

- **Registro de imágenes del Servicio de control de Astra**: Utilice esta opción si no utiliza un registro local con las imágenes del Centro de control de Astra o si prefiere este método a la descarga del paquete desde el Sitio de soporte de NetApp.
- **Sitio de soporte de NetApp**: Utilice esta opción si utiliza un registro local con las imágenes del Centro de control de Astra.

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.

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.

   ```
   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
   ```

Complete los pasos adicionales si utiliza un registro local

Si tiene pensado enviar el paquete Centro de control de Astra a su registro local, debe usar el complemento de la línea de comandos kubectl de Astra de NetApp.

Instale el complemento Astra kubectl de NetApp

Complete estos pasos para instalar el plugin de línea de comandos kubectl de NetApp Astra más reciente.

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 a su registro

1. Si planeas enviar el paquete Astra Control Center a tu registro local, completa la secuencia de pasos apropiada para tu motor de contenedores:

Docker

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

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

b. 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

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

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

b. Inicie sesión en su registro:

```
podman login <YOUR_REGISTRY>
```

c. 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=24.02.0-69
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=24.02.0-69
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/24.02.0-69/image:version
```

2. Cambie el directorio:

```
cd manifests
```

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:

   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/24.02.0-69`).

```
kubectl create secret docker-registry astra-registry-cred -n netapp-acc-
operator --docker-server=cr.astra.netapp.io --docker
-username=[astra_account_id] --docker-password=[astra_api_token]
```

+

```
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.

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

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

b. Cree un secreto para `netapp-acc` (o nombre personalizado). Agregue información de Docker y ejecute uno de los comandos adecuados en función de sus preferencias de registro:

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-
acc or custom namespace] --docker-server=cr.astra.netapp.io --docker
-username=[astra_account_id] --docker-password=[astra_api_token]
```

```
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. (Solo registros locales) Si está utilizando un registro local, complete estos pasos:

   a. Abra el YAML de implementación del operador de Astra Control Center:

   ```
   vim astra_control_center_operator_deploy.yaml
   ```

   Una muestra anotada de AYLMA sigue estos pasos.

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

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

   c. 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.

   d. 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.

```
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:24.02.68
        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
```

2. 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
```

3. 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:

Nombre de cuenta

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`

Versión astraVersion

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	`24.02.0-69`

Dirección de astern

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`

AutoSupport

Sus selecciones en esta sección determinan si participará en la aplicación de soporte proactivo de NetApp, NetApp Active IQ 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`

correo electrónico

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`

Nombre

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`

Apellidos

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`

ImageRegistry

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 de Astra Control, que aloja todas las imágenes necesarias para implementar Astra Control Center. El valor se rellenará previamente y no será necesario realizar ninguna acción a menos que haya configurado un registro local. Para un registro local, reemplace este valor existente por el nombre del registro de imágenes donde insertó las imágenes en el paso anterior. No utilizar `http://` o. `https://` en el nombre del registro.	cadena	`cr.astra.netapp.io` (predeterminado) `example.registry.com/astra` (ejemplo de registro local)

Ajuste	Uso	Orientación	Tipo	Ejemplo
`imageRegistry.secret`	Opcional	El nombre del secreto Kubernetes utilizado para autenticarse con el registro de imágenes. El valor se rellenará previamente y no será necesario realizar ninguna acción a menos que haya configurado un registro local y la cadena que haya introducido para ese registro en `imageRegistry.name` requiere un secreto. IMPORTANTE: Si está utilizando un registro local que no requiere autorización, debe eliminarlo `secret` línea dentro `imageRegistry` o se producirá un error en la instalación.	cadena	`astra-registry-cred`

Clase de almacenamiento

Ajuste	Orientación	Tipo	Ejemplo
`storageClass`	Cambie el `storageClass` valor desde `ontap-gold` A otro recurso de Storage Class según lo requiera la instalación. Ejecute el comando `kubectl get sc` para determinar las clases de almacenamiento configuradas existentes. Una de las clases de almacenamiento configuradas por el aprovisionador de Astra Control debe introducirse 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`

VolumeReclaimPolicy

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`

IngresType

Ajuste	Orientación	Tipo	Opciones
`ingressType`	Utilice uno de los siguientes tipos de entrada: **Genérico** (`ingressType: "Generic"`) (Predeterminado) Utilice esta opción cuando tenga otro controlador de entrada en uso o prefiera utilizar su propio controlador de entrada. Después de poner en marcha Astra Control Center, será necesario 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`

Tamaño escalonado

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)

Recursos astrarScaler

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`

Valores adicionales

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
```

crds

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.

muestra astrara_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: "[cr.astra.netapp.io or 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

```
acc-helm-repo-5bd77c9ddd-8wxm2             1/1     Running     0
1h
activity-5bb474dc67-8l9ss                  1/1     Running     0
1h
activity-5bb474dc67-qbrtq                  1/1     Running     0
1h
api-token-authentication-6wbj2             1/1     Running     0
1h
api-token-authentication-9pgw6             1/1     Running     0
1h
api-token-authentication-tqf6d             1/1     Running     0
1h
asup-5495f44dbd-z4kft                      1/1     Running     0
1h
authentication-6fdd899858-5x45s            1/1     Running     0
1h
bucketservice-84d47487d-n9xgp              1/1     Running     0
1h
bucketservice-84d47487d-t5jhm              1/1     Running     0
1h
cert-manager-5dcb7648c4-hbldc              1/1     Running     0
1h
cert-manager-5dcb7648c4-nr9qf              1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-bk2tf   1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-pfnck   1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-ngz2x       1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-rwtbn       1/1     Running     0
1h
certificates-5f5b7b4dd-52tnj               1/1     Running     0
1h
certificates-5f5b7b4dd-gtjbx               1/1     Running     0
1h
certificates-expiry-check-28477260-dz5vw   0/1     Completed   0
1h
cloud-extension-6f58cc579c-lzfmv           1/1     Running     0
1h
cloud-extension-6f58cc579c-zw2km           1/1     Running     0
1h
cluster-orchestrator-79dd5c8d95-qjg92      1/1     Running     0
1h
```

```
composite-compute-85dc84579c-nz82f         1/1     Running     0
1h
composite-compute-85dc84579c-wx2z2         1/1     Running     0
1h
composite-volume-bff6f4f76-789nj           1/1     Running     0
1h
composite-volume-bff6f4f76-kwnd4           1/1     Running     0
1h
credentials-79fd64f788-m7m8f               1/1     Running     0
1h
credentials-79fd64f788-qnc6c               1/1     Running     0
1h
entitlement-f69cdbd77-4p2kn                1/1     Running     0
1h
entitlement-f69cdbd77-hswm6                1/1     Running     0
1h
features-7b9585444c-7xd7m                  1/1     Running     0
1h
features-7b9585444c-dcqwc                  1/1     Running     0
1h
fluent-bit-ds-crq8m                        1/1     Running     0
1h
fluent-bit-ds-gmgq8                        1/1     Running     0
1h
fluent-bit-ds-gzr4f                        1/1     Running     0
1h
fluent-bit-ds-j6sf6                        1/1     Running     0
1h
fluent-bit-ds-v4t9f                        1/1     Running     0
1h
fluent-bit-ds-x7j59                        1/1     Running     0
1h
graphql-server-6cc684fb46-2x8lr            1/1     Running     0
1h
graphql-server-6cc684fb46-bshbd            1/1     Running     0
1h
hybridauth-84599f79fd-fjc7k                1/1     Running     0
1h
hybridauth-84599f79fd-s9pmn                1/1     Running     0
1h
identity-95df98cb5-dvlmz                   1/1     Running     0
1h
identity-95df98cb5-krf59                   1/1     Running     0
1h
influxdb2-0                                1/1     Running     0
1h
```

```
keycloak-operator-6d4d688697-cfq8b         1/1     Running     0
1h
krakend-5d5c8f4668-7bq8g                   1/1     Running     0
1h
krakend-5d5c8f4668-t8hbn                   1/1     Running     0
1h
license-689cdd4595-2gsc8                   1/1     Running     0
1h
license-689cdd4595-g6vwk                   1/1     Running     0
1h
login-ui-57bb599956-4fwgz                  1/1     Running     0
1h
login-ui-57bb599956-rhztb                  1/1     Running     0
1h
loki-0                                     1/1     Running     0
1h
metrics-facade-846999bdd4-f7jdm            1/1     Running     0
1h
metrics-facade-846999bdd4-lnsxl            1/1     Running     0
1h
monitoring-operator-6c9d6c4b8c-ggkrl       2/2     Running     0
1h
nats-0                                     1/1     Running     0
1h
nats-1                                     1/1     Running     0
1h
nats-2                                     1/1     Running     0
1h
natssync-server-6df7d6cc68-9v2gd           1/1     Running     0
1h
nautilus-64b7fbdd98-bsgwb                  1/1     Running     0
1h
nautilus-64b7fbdd98-djlhw                  1/1     Running     0
1h
openapi-864584bccc-75nlv                   1/1     Running     0
1h
openapi-864584bccc-zh6bx                   1/1     Running     0
1h
polaris-consul-consul-server-0             1/1     Running     0
1h
polaris-consul-consul-server-1             1/1     Running     0
1h
polaris-consul-consul-server-2             1/1     Running     0
1h
polaris-keycloak-0                         1/1     Running     2 (1h
ago)      1h
```

```
polaris-keycloak-1                          1/1     Running     0
1h
polaris-keycloak-db-0                       1/1     Running     0
1h
polaris-keycloak-db-1                       1/1     Running     0
1h
polaris-keycloak-db-2                       1/1     Running     0
1h
polaris-mongodb-0                           1/1     Running     0
1h
polaris-mongodb-1                           1/1     Running     0
1h
polaris-mongodb-2                           1/1     Running     0
1h
polaris-ui-66476dcf87-f6s8j                 1/1     Running     0
1h
polaris-ui-66476dcf87-ztjk7                 1/1     Running     0
1h
polaris-vault-0                             1/1     Running     0
1h
polaris-vault-1                             1/1     Running     0
1h
polaris-vault-2                             1/1     Running     0
1h
public-metrics-bfc4fc964-x4m79              1/1     Running     0
1h
storage-backend-metrics-7dbb88d4bc-g78cj    1/1     Running     0
1h
storage-provider-5969b5df5-hjvcm            1/1     Running     0
1h
storage-provider-5969b5df5-r79ld            1/1     Running     0
1h
task-service-5fc9dc8d99-4q4f4               1/1     Running     0
1h
task-service-5fc9dc8d99-8l5zl               1/1     Running     0
1h
task-service-task-purge-28485735-fdzkd      1/1     Running     0
12m
telegraf-ds-2rgm4                           1/1     Running     0
1h
telegraf-ds-4qp6r                           1/1     Running     0
1h
telegraf-ds-77frs                           1/1     Running     0
1h
telegraf-ds-bc725                           1/1     Running     0
1h
```

```
telegraf-ds-cvmxf                          1/1     Running     0
1h
telegraf-ds-tqzgj                          1/1     Running     0
1h
telegraf-rs-5wtd8                          1/1     Running     0
1h
telemetry-service-6747866474-5djnc         1/1     Running     0
1h
telemetry-service-6747866474-thb7r         1/1     Running     1 (1h
ago)      1h
tenancy-5669854fb6-gzdzf                   1/1     Running     0
1h
tenancy-5669854fb6-xvsm2                   1/1     Running     0
1h
traefik-8f55f7d5d-4lgfw                    1/1     Running     0
1h
traefik-8f55f7d5d-j4wt6                    1/1     Running     0
1h
traefik-8f55f7d5d-p6gcq                    1/1     Running     0
1h
trident-svc-7cb5bb4685-54cnq               1/1     Running     0
1h
trident-svc-7cb5bb4685-b28xh               1/1     Running     0
1h
vault-controller-777b9bbf88-b5bqt          1/1     Running     0
1h
vault-controller-777b9bbf88-fdfd8          1/1     Running     0
1h
```

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 obtén la contraseña de configuración inicial que usarás cuando inicies 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  24.02.0-69
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 poner en marcha Astra Control Center, deberás configurar la controladora de entrada para exponer Astra Control Center con una 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

Tras instalar Astra Control Center, cambiará la contraseña del administrador predeterminado e iniciará sesión en la consola 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
```

Procedimientos de instalación alternativos

- **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.

El futuro

- (Opcional) en función de su entorno, post-instalación completa "pasos de configuración".
- "Después de instalar Astra Control Center, iniciar sesión en la interfaz de usuario y cambiar la contraseña, querrá configurar una licencia, añadir clústeres, habilitar la autenticación, gestionar el almacenamiento y añadir buckets".

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:

```yaml
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.