# Instalar Astra Control Center

## Astra Control Center

NetApp
June 06, 2024

# Tabla de contenidos

# 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 para instalar Astra Control Center Operator y Astra Control Center en su entorno. Puede utilizar este procedimiento para instalar Astra Control Center en entornos conectados a Internet o con conexión por aire.

Para entornos Red Hat OpenShift, puede utilizar un "procedimiento alternativo" Para instalar Astra Control Center con OpenShift OperatorHub.

**Lo que necesitará**

- "Antes de comenzar la instalación, prepare su entorno para la implementación de Astra Control Center".
- Si ha configurado o desea configurar directivas de seguridad de POD en su entorno, familiarícese con las directivas de seguridad de POD y cómo afectan a la instalación de Astra Control Center. Consulte "Comprender las restricciones de directivas de seguridad de POD".
- Asegurarse de que todos los operadores del clúster se encuentren en estado correcto y estén disponibles.

```
kubectl get clusteroperators
```

- Asegúrese de que todos los servicios de API se encuentren en buen estado y estén disponibles:

```
kubectl get apiservices
```

- Asegúrese de que el FQDN de Astra que tiene previsto utilizar se puede enrutar a este 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.
- Si ya existe un administrador de certificados en el clúster, tendrá que realizar algunos "requisitos previos" Por lo tanto, Astra Control Center no instala su propio cert-Manager.

**Acerca de esta tarea**

El proceso de instalación de Astra Control Center realiza lo siguiente:

- Instala los componentes de Astra en `netapp-acc` (o nombre personalizado).
- Crea una cuenta predeterminada.
- Establece una dirección de correo electrónico de usuario administrativo predeterminada y una contraseña única predeterminada. A este usuario se le asigna el rol de propietario del sistema que se necesita para iniciar sesión por primera vez en la interfaz de usuario.
- Le ayuda a determinar que se están ejecutando todas las pods de Astra Control Center.
- Instala la interfaz de usuario de Astra.

(Se aplica sólo a la versión Astra Data Store Early Access Program (EAP)) Si tiene intención de gestionar Astra Data Store mediante Astra Control Center y habilitar los flujos de trabajo de VMware, implemente Astra Control Center únicamente en `pcloud` espacio de nombres y no en `netapp-acc` espacio de nombres o un espacio de nombres personalizado que se describe en los pasos de este procedimiento.

No ejecute el siguiente comando durante todo el proceso de instalación para evitar eliminar todas las POD de Astra Control Center: `kubectl delete -f astra_control_center_operator_deploy.yaml`

Si utiliza Podman de Red Hat en lugar de Docker Engine, los comandos de Podman se pueden utilizar en lugar de los comandos de Docker.

**Pasos**

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

- Descargue y desembale el paquete 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

# Descargue y desembale el paquete Astra Control Center

1. Descargue el paquete Astra Control Center (`astra-control-center-[version].tar.gz`) del "Sitio de soporte de NetApp".
2. Descargue el archivo zip de los certificados y claves de Astra Control Center de "Sitio de soporte de NetApp".
3. (Opcional) Use el siguiente comando para verificar la firma del paquete:

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

4. Extraiga las imágenes:

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

# Instale el complemento Astra kubectl de NetApp

La Astra de NetApp `kubectl` El complemento de línea de comandos ahorra tiempo al realizar tareas comunes asociadas con la implementación y actualización de Astra Control Center.

**Lo que necesitará**

NetApp proporciona binarios para el complemento para distintas arquitecturas de CPU y sistemas operativos. Debe saber qué CPU y sistema operativo tiene antes de realizar esta tarea. En los sistemas operativos Linux y Mac, puede utilizar `uname -a` comando para recopilar esta información.

**Pasos**

1. Enumere la Astra de NetApp disponible `kubectl` Haga un complemento para binarios y anote el nombre del archivo que necesita para su sistema operativo y arquitectura de CPU:

```
ls kubectl-astra/
```

2. Copie el archivo en la misma ubicación que el estándar `kubectl` utilidad. En este ejemplo, la `kubectl` la utilidad se encuentra en `/usr/local/bin` directorio. Sustituya `<binary-name>` con el nombre del archivo que necesita:

```
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 Astra:

```
cd acc
```

2. Push las imágenes del paquete del directorio imagen de Astra Control Center en su registro local. Realice las siguientes sustituciones antes de ejecutar el comando:
   - Sustituya BUNDLE_FILE por el nombre del archivo Astra Control Bundle (por ejemplo, `acc.manifest.yaml`).
   - Sustituya MY_REGISTRATION por la URL del repositorio de Docker.
   - Sustituya MY_REGISTRATION_USER por el nombre de usuario.
   - Sustituya MY_REGISTRATION_TOKEN por un token autorizado para el registro.

```
kubectl astra packages push-images -m BUNDLE_FILE -r MY_REGISTRY
-u MY_REGISTRY_USER -p MY_REGISTRY_TOKEN
```

**Podman**

1. Inicie sesión en su registro:

```
podman login [your_registry_path]
```

2. Ejecute el siguiente script, haciendo la sustitución de <YOUR_REGISTRY> como se indica en los comentarios:

```
# You need to be at the root of the tarball.
# You should see these files to confirm correct location:
#   acc.manifest.yaml
#   acc/

# Replace <YOUR_REGISTRY> with your own registry (e.g
registry.customer.com or registry.customer.com/testing, etc..)
export REGISTRY=<YOUR_REGISTRY>
export PACKAGENAME=acc
export PACKAGEVERSION=22.08.1-26
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
  # Load to local cache
  astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image(s): //')

  # Remove path and keep imageName.
  astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')

  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/netapp/astra/${PACKAGENAME}
/${PACKAGEVERSION}/${astraImageNoPath}

  # Push to the local repo.
  podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

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

1. Exporte el KUBECONFIG para el clúster de host de Astra Control Center:

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

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
```

   Respuesta:

```
namespace/netapp-acc-operator created
```

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/22.08.1-26`).

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

Respuesta de ejemplo:

```
secret/astra-registry-cred created
```

> Si elimina el espacio de nombres después de que se genere el secreto, deberá volver a generar el secreto para el espacio de nombres después de volver a crear el espacio de nombres.

c. Cree el `netapp-acc` (o espacio de nombres personalizado).

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

Respuesta de ejemplo:

```
namespace/netapp-acc created
```

d. Cree un secreto para `netapp-acc` (o espacio de nombres 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]
```

Respuesta

```
secret/astra-registry-cred created
```

a. [[substep_kubeconfig_secret](opcional) Si desea que el clúster sea gestionado automáticamente por Astra Control Center después de la instalación, asegúrese de proporcionar el kubeconfig como secreto dentro del espacio de nombres Astra Control Center que tiene intención de implementar utilizando este comando:

```
kubectl create secret generic [acc-kubeconfig-cred or custom secret name] --from-file=<path-to-your-kubeconfig> -n [netapp-acc or custom namespace]
```

# 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 `[your_registry_path]` para la `kube-rbac-proxy` imagen a la ruta del registro en la que se insertó la imagen en un paso anterior.

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

d. (Para instalaciones que utilizan la vista previa de Astra Data Store) Consulte este problema conocido con respecto a "Los aprovisionadores de clases de almacenamiento y los cambios adicionales que deberá realizar en la 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
  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: [your_registry_path]/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
        command:
        - /manager
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        image: [your_registry_path]/acc-operator:[version x.y.z]
        imagePullPolicy: IfNotPresent
      imagePullSecrets: []
```

3. Instale el operador de Astra Control Center:

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

Respuesta de ejemplo:

```
namespace/netapp-acc-operator created
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.astra.
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_min.yaml`) Para realizar las configuraciones de cuenta, AutoSupport, Registro y otras necesarias:

> `astra_control_center_min.yaml` Es la CR predeterminada y es adecuada para la mayoría de las instalaciones. Familiarícese con todos "Opciones CR y sus valores potenciales" Garantizar la puesta en marcha correcta de Astra Control Center en su entorno. Si se requieren personalizaciones adicionales para su entorno, puede utilizar `astra_control_center.yaml` Como CR alternativo.

```
vim astra_control_center_min.yaml
```

> Si está utilizando un registro que no requiere autorización, debe eliminar `secret` línea dentro `imageRegistry` o se producirá un error en la instalación.

a. Cambiar `[your_registry_path]` a la ruta de acceso del registro en la que ha insertado las imágenes en el paso anterior.

b. Cambie el `accountName` cadena al nombre que desea asociar a la cuenta.

c. Cambie el `astraAddress` Cadena al FQDN que desea utilizar en su navegador para acceder a Astra. No utilizar `http://` o. `https://` en la dirección. Copie este FQDN para utilizarlo en un paso posterior.

d. Cambie el `email` cadena en la dirección inicial predeterminada del administrador. Copie esta dirección de correo electrónico para su uso en un paso posterior.

e. Cambiar `enrolled` Para AutoSupport a. `false` para sitios sin conexión a internet o retención `true` para sitios conectados.

f. Si utiliza un administrador de certificados externo, añada las siguientes líneas a. `spec`:

```
spec:
  crds:
    externalCertManager: true
```

g. (Opcional) Añada un nombre `firstName` y apellidos `lastName` del usuario asociado con la cuenta. Este paso se puede realizar ahora o una versión posterior dentro de la interfaz de usuario.

h. (Opcional) cambie el `storageClass` Valor en otro recurso de la clase de almacenamiento de Trident, si es necesario para su instalación.

i. (Opcional) Si desea que el clúster sea gestionado automáticamente por Astra Control Center después de la instalación y ya lo tiene se ha creado el secreto que contiene el kubeconfig para este cluster, Proporcione el nombre del secreto agregando un nuevo campo a este archivo YLMA llamado `astraKubeConfigSecret: "acc-kubeconfig-cred or custom secret name"`

j. Realice uno de los siguientes pasos:

- **Otro controlador de entrada (ingressType:Generic)**: Esta es la acción predeterminada con Astra Control Center. Después de implementar Astra Control Center, deberá configurar el controlador Ingress para exponer Astra Control Center con una dirección URL.

  La instalación predeterminada de Astra Control Center configura su puerta de enlace (`service/traefik`) ser del tipo `ClusterIP`. Esta instalación predeterminada requiere que configure además un dispositivo de entrada/controlador de Kubernetes para enrutar el tráfico hacia él. Si desea utilizar una entrada, consulte "Configure la entrada para el equilibrio de carga".

- **Equilibrador de carga de servicio (ingressType:AccTraefik)**: Si no desea instalar un controlador IngressController o crear un recurso de entrada, establezca `ingressType` para `AccTraefik`.

  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 asigne una dirección IP externa accesible. Si se permiten equilibradores de carga en su entorno y 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 más información sobre el tipo de servicio de "LoadBalancer" y la entrada, consulte "Requisitos".

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  astraKubeConfigSecret: "acc-kubeconfig-cred or custom secret name"
  ingressType: "Generic"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
```

# 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]
   ```

   Respuesta de ejemplo:

   ```
   namespace/netapp-acc created
   ```

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

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

   Respuesta de ejemplo:

   ```
   astracontrolcenter.astra.netapp.io/astra created
   ```

# Comprobar el estado del sistema

Si prefiere utilizar OpenShift, puede utilizar comandos de OC comparables para realizar los pasos de verificación.

1. 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.

**Ejemplo de respuesta**

```
NAME                                        READY   STATUS    RESTARTS
AGE
acc-helm-repo-6b44d68d94-d8m55              1/1     Running   0
13m
activity-78f99ddf8-hltct                    1/1     Running   0
10m
api-token-authentication-457nl              1/1     Running   0
9m28s
api-token-authentication-dgwsz              1/1     Running   0
9m28s
api-token-authentication-hmqqc              1/1     Running   0
9m28s
asup-75fd554dc6-m6qzh                       1/1     Running   0
9m38s
authentication-6779b4c85d-92gds             1/1     Running   0
8m11s
bucketservice-7cc767f8f8-lqwr8              1/1     Running   0
9m31s
certificates-549fd5d6cb-5kmd6               1/1     Running   0
9m56s
certificates-549fd5d6cb-bkjh9               1/1     Running   0
9m56s
cloud-extension-7bcb7948b-hn8h2             1/1     Running   0
10m
cloud-insights-service-56ccf86647-fgg69     1/1     Running   0
9m46s
composite-compute-677685b9bb-7vgsf          1/1     Running   0
10m
composite-volume-657d6c5585-dnq79           1/1     Running   0
9m49s
credentials-755fd867c8-vrlmt                1/1     Running   0
11m
entitlement-86495cdf5b-nwhh2                1/1     Running   2
10m
features-5684fb8b56-8d6s8                   1/1     Running   0
10m
fluent-bit-ds-rhx7v                         1/1     Running   0
7m48s
fluent-bit-ds-rjms4                         1/1     Running   0
7m48s
fluent-bit-ds-zf5ph                         1/1     Running   0
7m48s
graphql-server-66d895f544-w6hjd             1/1     Running   0
3m29s
```

```
identity-744df448d5-rlcmm                1/1    Running  0
10m
influxdb2-0                              1/1    Running  0
13m
keycloak-operator-75c965cc54-z7csw       1/1    Running  0
8m16s
krakend-798d6df96f-9z2sk                 1/1    Running  0
3m26s
license-5fb7d75765-f8mjg                 1/1    Running  0
9m50s
login-ui-7d5b7df85d-l2s7s                1/1    Running  0
3m20s
loki-0                                   1/1    Running  0
13m
metrics-facade-599b9d7fcc-gtmgl          1/1    Running  0
9m40s
monitoring-operator-67cc74f844-cdplp     2/2    Running  0
8m11s
nats-0                                   1/1    Running  0
13m
nats-1                                   1/1    Running  0
13m
nats-2                                   1/1    Running  0
12m
nautilus-769f5b74cd-k5jxm                1/1    Running  0
9m42s
nautilus-769f5b74cd-kd9gd                1/1    Running  0
8m59s
openapi-84f6ccd8ff-76kvp                 1/1    Running  0
9m34s
packages-6f59fc67dc-4g2f5                1/1    Running  0
9m52s
polaris-consul-consul-server-0           1/1    Running  0
13m
polaris-consul-consul-server-1           1/1    Running  0
13m
polaris-consul-consul-server-2           1/1    Running  0
13m
polaris-keycloak-0                       1/1    Running  0
8m7s
polaris-keycloak-1                       1/1    Running  0
5m49s
polaris-keycloak-2                       1/1    Running  0
5m15s
polaris-keycloak-db-0                    1/1    Running  0
8m6s
```

```
polaris-keycloak-db-1                     1/1    Running  0
5m49s
polaris-keycloak-db-2                     1/1    Running  0
4m57s
polaris-mongodb-0                         2/2    Running  0
13m
polaris-mongodb-1                         2/2    Running  0
12m
polaris-mongodb-2                         2/2    Running  0
12m
polaris-ui-565f56bf7b-zwr8b               1/1    Running  0
3m19s
polaris-vault-0                           1/1    Running  0
13m
polaris-vault-1                           1/1    Running  0
13m
polaris-vault-2                           1/1    Running  0
13m
public-metrics-6d86d66444-2wbzl           1/1    Running  0
9m30s
storage-backend-metrics-77c5d98dcd-dbhg5 1/1    Running  0
9m44s
storage-provider-78c885f57c-6zcv4         1/1    Running  0
9m36s
telegraf-ds-2l2m9                         1/1    Running  0
7m48s
telegraf-ds-qfzgh                         1/1    Running  0
7m48s
telegraf-ds-shrms                         1/1    Running  0
7m48s
telegraf-rs-bjpkt                         1/1    Running  0
7m48s
telemetry-service-6684696c64-qzfdf        1/1    Running  0
10m
tenancy-6596b6c54d-vmpsm                  1/1    Running  0
10m
traefik-7489dc59f9-6mnst                  1/1    Running  0
3m19s
traefik-7489dc59f9-xrkgg                  1/1    Running  0
3m4s
trident-svc-6c8dc458f5-jswcl              1/1    Running  0
10m
vault-controller-6b954f9b76-gz9nm         1/1    Running  0
11m
```

2. (Opcional) para asegurarse de que la instalación ha finalizado, puede ver el `acc-operator` registra utilizando el siguiente comando.

```
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 un error de registro del clúster que se indica en los registros, puede volver a intentar el registro a través del flujo de trabajo de add cluster "En la interfaz de usuario de" O API.

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

```
kubectl get AstraControlCenter -n netapp-acc
```

Respuesta:

```
NAME    UUID                                   VERSION     ADDRESS
READY
astra   ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f  22.08.1-26
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 una controladora de entrada de Kubernetes que gestione el acceso externo a los servicios, como el equilibrio de carga en un clúster.

Este procedimiento explica cómo configurar un controlador de entrada (`ingressType:Generic`). Esta es la acción predeterminada con Astra Control Center. Después de implementar Astra Control Center, deberá configurar el controlador Ingress para exponer Astra Control Center con una dirección URL.

> Si no desea configurar un controlador de entrada, puede configurarlo `ingressType:AccTraefik`). 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 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 más información sobre el tipo de servicio de "LoadBalancer" y la entrada, consulte "Requisitos".

Los pasos varían en función del tipo de controlador de entrada que utilice:

- Entrada Istio
- Controlador de entrada nginx
- Controlador OpenShift Ingress

**Lo que necesitará**

- El requerido "controlador de entrada" ya debe ponerse en marcha.
- La "clase de entrada" ya se debe crear la correspondiente al controlador de entrada.
- Se utilizan versiones de Kubernetes entre e incluidas v1.19 y v1.22.

**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 uso del tipo de recurso v1beta1 (obsoleto en la versión de Kubernetes menor que o 1.22) o v1 para un esquema obsoleto o nuevo:

   Salida:

```
apiVersion: networking.k8s.io/v1beta1
kind: IngressClass
metadata:
  name: istio
spec:
  controller: istio.io/ingress-controller
---
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress
  namespace: istio-system
spec:
  ingressClassName: istio
  tls:
  - hosts:
    - <ACC addess>
    secretName: [tls secret name]
  rules:
  - host: [ACC addess]
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          serviceName: traefik
          servicePort: 80
```

Para el nuevo esquema v1, siga este ejemplo:

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

Salida:

```
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: istio-system
spec:
  ingressClassName: istio
  tls:
  - hosts:
    - <ACC addess>
    secretName: [tls secret name]
  rules:
  - host: [ACC addess]
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: traefik
            port:
              number: 80
```

5. Implementar Astra Control Center como es habitual.
6. Compruebe el estado de la entrada:

```
kubectl get ingress -n netapp-acc
```

Respuesta:

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

**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` (o nombre personalizado) mediante el `v1beta1` (Obsoleto en la versión de Kubernetes inferior a o 1.22) o. `v1` tipo de recurso para un esquema obsoleto o nuevo:

   a. Para un `v1beta1` esquema obsoleto, siga este ejemplo:

```
apiVersion: extensions/v1beta1
Kind: IngressClass
metadata:
  name: ingress-acc
  namespace: [netapp-acc or custom namespace]
  annotations:
    kubernetes.io/ingress.class: [class name for nginx controller]
spec:
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - backend:
        serviceName: traefik
        servicePort: 80
        pathType: ImplementationSpecific
```

   b. Para la `v1` nuevo esquema, siga 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 addess>
    http:
      paths:
        - path:
          backend:
            service:
              name: traefik
              port:
                number: 80
          pathType: ImplementationSpecific
```

**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 explorador, introduzca el FQDN que utilizó en `astraAddress` en la `astra_control_center_min.yaml` CR cuando Ha instalado Astra Control Center.

2. Acepte los certificados autofirmados cuando se le solicite.

   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_min.yaml` CR cuando Ha instalado Astra Control Center, seguido de la contraseña única (`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 este es su primer inicio de sesión y olvida la contraseña y aún no se han creado otras cuentas de usuario administrativas, comuníquese con el servicio de 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.

**Pasos**

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

```
kubectl logs --follow -n netapp-acc-operator $(kubectl get pods -n
netapp-acc-operator -o name) -c manager
```

# El futuro

Complete la implementación llevando a cabo "tareas de configuración".

=
:allow-uri-read:

# Comprender las restricciones de directivas de seguridad de POD

Astra Control Center admite la limitación de privilegios mediante directivas de seguridad de POD (PSP). Las políticas de seguridad de POD permiten limitar los usuarios o grupos que pueden ejecutar contenedores y los privilegios que dichos contenedores pueden tener.

Algunas distribuciones de Kubernetes, como RKE2, tienen una política de seguridad de POD predeterminada

que es demasiado restrictiva y provoca problemas al instalar Astra Control Center.

Puede utilizar la información y los ejemplos que se incluyen aquí para comprender las directivas de seguridad de POD que Astra Control Center crea y configurar las directivas de seguridad de POD que proporcionan la protección necesaria sin interferir con las funciones de Astra Control Center.

## PSP instalado por Astra Control Center

Astra Control Center crea varias directivas de seguridad de POD durante la instalación. Algunas de ellas son permanentes y algunas se crean durante ciertas operaciones y se eliminan una vez que se completa la operación.

#### Se crean PSP durante la instalación

Durante la instalación de Astra Control Center, el operador Astra Control Center instala una directiva de seguridad de POD personalizada, un objeto Role y un objeto RoleBinding para admitir la implementación de los servicios Astra Control Center en el espacio de nombres Astra Control Center.

La política y los objetos nuevos tienen los siguientes atributos:

```
kubectl get psp

NAME                              PRIV    CAPS          SELINUX    RUNASUSER
FSGROUP     SUPGROUP    READONLYROOTFS   VOLUMES
avp-psp                           false                 RunAsAny   RunAsAny
RunAsAny    RunAsAny    false            *
netapp-astra-deployment-psp    false                 RunAsAny   RunAsAny
RunAsAny    RunAsAny    false            *

kubectl get role

NAME                                     CREATED AT
netapp-astra-deployment-role             2022-06-27T19:34:58Z

kubectl get rolebinding

NAME                                     ROLE
AGE
netapp-astra-deployment-rb               Role/netapp-astra-deployment-role
32m
```

#### Se crean PSP durante las operaciones de backup

Durante las operaciones de copia de seguridad, Astra Control Center crea una política de seguridad de POD dinámica, un objeto ClusterRole y un objeto RoleBinding. Estos permiten utilizar el proceso de backup, que se produce en un espacio de nombres aparte.

La política y los objetos nuevos tienen los siguientes atributos:

```
kubectl get psp

NAME                             PRIV    CAPS
SELINUX    RUNASUSER          FSGROUP     SUPGROUP    READONLYROOTFS
VOLUMES
netapp-astra-backup             false   DAC_READ_SEARCH
RunAsAny   RunAsAny           RunAsAny    RunAsAny    false            *

kubectl get role

NAME                   CREATED AT
netapp-astra-backup   2022-07-21T00:00:00Z

kubectl get rolebinding

NAME                   ROLE                          AGE
netapp-astra-backup   Role/netapp-astra-backup   62s
```

**Se crean PSP durante la gestión del clúster**

Cuando gestiona un clúster, Astra Control Center instala el operador de supervisión de netapp en el clúster gestionado. Este operador crea una directiva de seguridad de POD, un objeto ClusterRole y un objeto RoleBinding para implementar servicios de telemetría en el espacio de nombres Astra Control Center.

La política y los objetos nuevos tienen los siguientes atributos:

```
kubectl get psp

NAME                             PRIV    CAPS
SELINUX    RUNASUSER          FSGROUP     SUPGROUP    READONLYROOTFS
VOLUMES
netapp-monitoring-psp-nkmo     true    AUDIT_WRITE,NET_ADMIN,NET_RAW
RunAsAny   RunAsAny           RunAsAny    RunAsAny    false            *

kubectl get role

NAME                                          CREATED AT
netapp-monitoring-role-privileged             2022-07-21T00:00:00Z

kubectl get rolebinding

NAME                                                ROLE
AGE
netapp-monitoring-role-binding-privileged           Role/netapp-
monitoring-role-privileged             2m5s
```

## Active la comunicación de red entre espacios de nombres

Algunos entornos utilizan construcciones de NetworkPolicy para restringir el tráfico entre espacios de nombres. El operador Astra Control Center, Astra Control Center y el complemento Astra para VMware vSphere están todos en espacios de nombres diferentes. Los servicios de estos distintos espacios de nombres deben poder comunicarse entre sí. Para activar esta comunicación, siga estos pasos.

**Pasos**

1. Elimine los recursos de NetworkPolicy que existan en el espacio de nombres de Astra Control Center:

```
kubectl get networkpolicy -n netapp-acc
```

2. Para cada objeto NetworkPolicy devuelto por el comando anterior, utilice el siguiente comando para eliminarlo. Sustituya <OBJECT_NAME> por el nombre del objeto devuelto:

```
kubectl delete networkpolicy <OBJECT_NAME> -n netapp-acc
```

3. Aplique el siguiente archivo de recursos para configurar el objeto de política de red ACC-avp con el fin de permitir que los servicios de Astra Plugin para VMware vSphere puedan realizar solicitudes a los servicios de Astra Control Center. Reemplace la información entre paréntesis <> por la información de su entorno:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: acc-avp-network-policy
  namespace: <ACC_NAMESPACE_NAME> # REPLACE THIS WITH THE ASTRA CONTROL
CENTER NAMESPACE NAME
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              kubernetes.io/metadata.name: <PLUGIN_NAMESPACE_NAME> #
REPLACE THIS WITH THE ASTRA PLUGIN FOR VMWARE VSPHERE NAMESPACE NAME
```

4. Aplique el siguiente archivo de recursos para configurar el objeto de directiva de red-operador de ACC con el fin de permitir que el operador de Astra Control Center se comunique con los servicios de Astra Control Center. Reemplace la información entre paréntesis <> por la información de su entorno:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: acc-operator-network-policy
  namespace: <ACC_NAMESPACE_NAME> # REPLACE THIS WITH THE ASTRA CONTROL
CENTER NAMESPACE NAME
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              kubernetes.io/metadata.name: <NETAPP-ACC-OPERATOR> #
REPLACE THIS WITH THE OPERATOR NAMESPACE NAME
```

## 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. Debe eliminarlos 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 el espacio de nombres ACC-netapp:

```
kubectl get quota -n netapp-acc
```

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
```

```
kubectl delete resourcequota  pods-low -n netapp-acc
```

```
kubectl delete resourcequota  pods-medium -n netapp-acc
```

3. Obtenga los rangos de límites en el espacio de nombres ACC-netapp:

```
kubectl get limits -n netapp-acc
```

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
```

=
:allow-uri-read:

## Información de copyright



## Información de la marca comercial