Manos a la obra

Astra Control Center

NetApp
August 11, 2025

Tabla de contenidos

Manos a la obra

Más información sobre Astra Control

Astra Control es una solución de gestión del ciclo de vida de los datos de las aplicaciones de Kubernetes que simplifica las operaciones para aplicaciones con estado. Proteja, cree backups, replique y migre cargas de trabajo de Kubernetes con facilidad y cree instantáneamente clones de aplicaciones en funcionamiento.

Funciones

Astra Control ofrece funcionalidades cruciales para la gestión del ciclo de vida de los datos de las aplicaciones Kubernetes:

- Gestione automáticamente el almacenamiento persistente
- Crear copias Snapshot y backups bajo demanda que se tienen en cuenta las aplicaciones
- Automatice las operaciones de backup y Snapshot condicionadas por políticas
- Migre aplicaciones y datos de un clúster de Kubernetes a otro
- Replicar aplicaciones en un sistema remoto mediante la tecnología SnapMirror de NetApp (Astra Control Center)
- Clone aplicaciones de almacenamiento provisional a producción
- Visualizar el estado de la protección y el estado de la aplicación
- Trabaje con una interfaz de usuario web o una API para implementar sus flujos de trabajo de backup y migración

Modelos de puesta en marcha

Astra Control está disponible en dos modelos de implementación:

- **Astra Control Service**: Un servicio gestionado por NetApp que proporciona gestión de datos para aplicaciones de clústeres de Kubernetes en varios entornos de proveedores de cloud, así como clústeres de Kubernetes autogestionados.
- **Astra Control Center**: Software autogestionado que proporciona gestión de datos para aplicaciones de clústeres de Kubernetes que se ejecutan en su entorno local. Astra Control Center también se puede instalar en entornos de varios proveedores de cloud con un entorno de administración del almacenamiento Cloud Volumes ONTAP de NetApp.

	Servicio de control Astra	Astra Control Center
¿Cómo se ofrece?	Como un servicio cloud totalmente gestionado de NetApp	Como software que se puede descargar, instalar y gestionar
¿Dónde está alojado?	En un cloud público que elija NetApp	En su propio clúster de Kubernetes
¿Cómo se actualiza?	Gestionado por NetApp	Usted administra cualquier actualización

	Servicio de control Astra	Astra Control Center
¿Cuáles son las distribuciones de Kubernetes compatibles?	• * Proveedores en la nube* ◦ Amazon Web Services ▪ Amazon Elastic Kubernetes Service (EKS) ◦ Google Cloud ▪ Google Kubernetes Engine (GKE) ◦ Microsoft Azure ▪ Azure Kubernetes Service (AKS) • **Clusters autogestionados** ◦ Kubernetes (ascendente) ◦ Motor Kubernetes de rancher (RKE) ◦ OpenShift Container Platform de Red Hat • * Clústeres locales* ◦ Red Hat OpenShift Container Platform en las instalaciones	• Azure Kubernetes Service en HCI de pila de Azure • Anthos de Google • Kubernetes (ascendente) • Motor Kubernetes de rancher (RKE) • OpenShift Container Platform de Red Hat

	Servicio de control Astra	Astra Control Center
¿Cuáles son los back-ends de almacenamiento compatibles?	• * Proveedores en la nube* ◦ Amazon Web Services ▪ Amazon EBS ▪ Amazon FSX para ONTAP de NetApp ▪ "Cloud Volumes ONTAP" ◦ Google Cloud ▪ Disco persistente de Google ▪ Cloud Volumes Service de NetApp ▪ "Cloud Volumes ONTAP" ◦ Microsoft Azure ▪ Discos gestionados de Azure ▪ Azure NetApp Files ▪ "Cloud Volumes ONTAP" • **Clusters autogestionados** ◦ Amazon EBS ◦ Discos gestionados de Azure ◦ Disco persistente de Google ◦ "Cloud Volumes ONTAP" ◦ NetApp MetroCluster ◦ "El Longhorn" • * Clústeres locales* ◦ NetApp MetroCluster ◦ Sistemas ONTAP AFF y FAS de NetApp ◦ ONTAP Select de NetApp ◦ "Cloud Volumes ONTAP" ◦ "El Longhorn"	• Sistemas ONTAP AFF y FAS de NetApp • ONTAP Select de NetApp • "Cloud Volumes ONTAP" • "El Longhorn"

Funcionamiento del servicio Astra Control

Astra Control Service es un servicio cloud gestionado por NetApp que siempre está activo y actualizado con las últimas funcionalidades. Utiliza varios componentes para habilitar la gestión del ciclo de vida de los datos de aplicaciones.

En un nivel superior, Astra Control Service funciona de esta manera:

- Para comenzar a trabajar con Astra Control Service, configure su proveedor de cloud y inscríbase para obtener una cuenta Astra.

    - Para los clústeres GKE, el servicio Astra Control utiliza "Cloud Volumes Service de NetApp para Google Cloud" O discos persistentes de Google como back-end de almacenamiento para sus volúmenes persistentes.
    - Para clústeres AKS, el servicio de control Astra utiliza "Azure NetApp Files" O Azure gestionó discos como back-end de almacenamiento para sus volúmenes persistentes.
    - Para clústeres de Amazon EKS, utiliza Astra Control Service "Amazon Elastic Block Store" o. "Amazon FSX para ONTAP de NetApp" como back-end de almacenamiento para sus volúmenes persistentes.
- Agregue su primera tecnología Kubernetes al servicio Astra Control. A continuación, el servicio de control de Astra realiza lo siguiente:
    - Crea un almacén de objetos en su cuenta de proveedor de cloud, que es donde se almacenan las copias de backup.

      En Azure, Astra Control Service también crea un grupo de recursos, una cuenta de almacenamiento y claves para el contenedor Blob.
    - Crea un nuevo rol de administrador y una cuenta de servicio de Kubernetes en el clúster.
    - Utiliza el nuevo rol de administrador para instalar el enlace../concepts/architecture#astra-control-components[Astra Control Provisioner^] en el clúster y crear una o varias clases de almacenamiento.
    - Si utilizas una oferta de almacenamiento de servicios en la nube de NetApp como back-end de almacenamiento, el servicio Astra Control utiliza el aprovisionador de control de Astra para aprovisionar volúmenes persistentes para tus aplicaciones. Si utiliza discos administrados de Amazon EBS o Azure como back-end de almacenamiento, deberá instalar un controlador CSI específico del proveedor. Se proporcionan instrucciones de instalación en "Configure Amazon Web Services" y.. "Configure Microsoft Azure con discos gestionados de Azure".
- En este momento, puede añadir aplicaciones al clúster. Se aprovisionan volúmenes persistentes en la nueva clase de almacenamiento predeterminada.
- A continuación, utilice Astra Control Service para gestionar estas aplicaciones y empiece a crear copias Snapshot, copias de seguridad y clones.

El plan gratuito de Astra Control le permite gestionar hasta 10 espacios de nombres en su cuenta. Si desea gestionar más de 10, deberá configurar la facturación actualizando del plan gratuito al plan Premium.

Cómo funciona Astra Control Center

Astra Control Center se ejecuta en forma local en su propia nube privada.

Astra Control Center admite los clústeres de Kubernetes con un tipo de almacenamiento configurado por el aprovisionador de Astra Control con un back-end de almacenamiento de ONTAP.

La supervisión y la telemetría limitadas (7 días de métricas) están disponibles en Astra Control Center y también se exportan a herramientas de supervisión nativas de Kubernetes (como Prometheus y Grafana) a través de puntos finales de métricas abiertas.

Astra Control Center está totalmente integrado en el ecosistema del asesor digital de AutoSupport y Active IQ (también conocido como asesor digital) para proporcionar a los usuarios y al servicio de soporte de NetApp información sobre solución de problemas e uso.

Puedes probar Astra Control Center con una licencia de evaluación integrada de 90 días. Mientras estás evaluando Astra Control Center, puedes obtener soporte a través del correo electrónico y las opciones de la comunidad. Además, tendrá acceso a los artículos de la base de conocimientos y a la documentación desde la consola de soporte del producto.

Para instalar y utilizar Astra Control Center, tendrá que estar seguro "requisitos".

En un nivel superior, Astra Control Center funciona de esta manera:

- Instala Astra Control Center en su entorno local. Obtenga más información sobre cómo "Instalar Astra Control Center".
- Puede realizar algunas tareas de configuración como las siguientes:
  - Configurar la licencia.
  - Añada el primer clúster.
  - Añada el back-end de almacenamiento que se detecta al añadir el clúster.
  - Agregue un bloque de almacenamiento de objetos que almacenará las copias de seguridad de la aplicación.

Obtenga más información sobre cómo "Configure Astra Control Center".

Puede añadir aplicaciones al clúster. O bien, si ya tiene algunas aplicaciones en el clúster que se están gestionando, puede utilizar Astra Control Center para gestionarlas. A continuación, utilice Astra Control Center para crear copias Snapshot, backups, clones y relaciones de replicación.

Si quiere más información

- "Documentación de Astra Control Service"
- "Documentación de Astra Control Center"
- "Documentación de Astra Trident"
- "Documentación de la API de Astra Control"
- "Documentación de ONTAP"

Requisitos del Centro de Control de Astra

Comience verificando la preparación de su entorno operativo, clústeres de aplicaciones, aplicaciones, licencias y explorador web. Asegúrate de que tu entorno cumpla con estos requisitos para poner en marcha y operar Astra Control Center.

Entornos de Kubernetes de clústeres host admitidos

Astra Control Center se ha validado con los siguientes entornos de host de Kubernetes:

> Compruebe que el entorno de Kubernetes que elijas para alojar Astra Control Center cumpla con los requisitos básicos de recursos que se describen en la documentación oficial del entorno.

Distribución de Kubernetes en clúster de hosts	Versiones compatibles
Azure Kubernetes Service en HCI de pila de Azure	Azure Stack HCI 21H2 y 22H2 con AKS 1.24.11 a 1.26.6
Anthos de Google	1,15 a 1,16 (consulte Requisitos de incorporación de Google Anthos)

Distribución de Kubernetes en clúster de hosts	Versiones compatibles
Kubernetes (ascendente)	1,27 a 1,29
Motor Kubernetes de rancher (RKE)	RKE 1: Versiones 1.24.17, 1.25.13, 1.26.8 con Rancher Manager 2.7.9 RKE 2: Versiones 1.23.16 y 1.24.13 con Rancher Manager 2.6.13 RKE 2: Versiones 1.24.17, 1.25.14, 1.26.9 con Rancher Manager 2.7.9
OpenShift Container Platform de Red Hat	4,12 hasta 4,14

Requisitos de recursos del clúster de hosts

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

> 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.

- **Extensiones de CPU**: Las CPU de todos los nodos del entorno de alojamiento deben tener habilitadas las extensiones AVX.
- * Nodos de trabajo*: Al menos 3 nodos de trabajo en total, con 4 núcleos de CPU y 12GB RAM cada uno
- **Requisitos de clúster de VMware Tanzu Kubernetes Grid**: Al alojar Astra Control Center en un clúster de VMware Tanzu Kubernetes Grid (TKG) o Tanzu Kubernetes Grid Integrated Edition (TKGi), tenga en cuenta las siguientes consideraciones.
  - El token predeterminado del archivo de configuración de VMware TKG y TKGi caduca diez horas después de la implementación. Si utiliza productos de la cartera de Tanzu, debe generar un archivo de configuración de tanzu Kubernetes Cluster con un token que no caduca para evitar problemas de conexión entre Astra Control Center y clústeres de aplicaciones administradas. Si desea obtener instrucciones, visite "La documentación de producto del centro de datos NSX-T de VMware."
  - Utilice la `kubectl get nsxlbmonitors -A` comando para ver si ya tiene un monitor de servicio configurado para aceptar tráfico de entrada. Si existe una, no debe instalar MetalLB, ya que el monitor de servicio existente anulará cualquier nueva configuración de equilibrador de carga.
  - Desactive la implementación predeterminada de la clase de almacenamiento TKG o TKGi en cualquier cluster de aplicaciones que Astra Control deba gestionar. Para ello, edite la `TanzuKubernetesCluster` recurso en el clúster de espacio de nombres.
  - Ten en cuenta los requisitos específicos para el aprovisionador de Astra Control al implementar Astra Control Center en un entorno TKG o TKGi:
    - El clúster debe admitir cargas de trabajo con privilegios.
    - La `--kubelet-dir` el indicador se debe establecer en la ubicación del directorio kubelet. De forma predeterminada, esta es `/var/vcap/data/kubelet`.
    - Especificación de la ubicación del kubelet mediante `--kubelet-dir` Sabe que funciona para el operador, Helm y. `tridentctl` implementaciones.

Requisitos de malla de servicio

Se recomienda instalar una versión vanilla compatible de la malla de servicio de Istio en el clúster de hosts de

Astra Control Center. Consulte "versiones compatibles" Para versiones compatibles de Istio. Los lanzamientos de marca de la malla de servicio de Istio, como OpenShift Service Mesh, no están validados con Astra Control Center.

Para integrar Astra Control Center con la malla de servicio de Istio instalada en el clúster de hosts, es necesario hacer la integración como parte de un Astra Control Center "instalación" y no independiente de este proceso.

> La instalación y el uso de Astra Control Center sin configurar una malla de servicio en el clúster de host tiene implicaciones de seguridad potencialmente graves.

Astra Trident

Si piensa utilizar Astra Trident en lugar de Astra Control Provisioner con esta versión, se admiten Astra Trident 23,04 y las versiones posteriores. Astra Control Center requerirá Aprovisionador de Astra Control en futuras versiones.

Aprovisionador de Astra Control

Para usar la funcionalidad de almacenamiento avanzada del aprovisionador de Astra Control, debe instalar Astra Trident 23,10 o una versión posterior y habilitarla "Funcionalidad de aprovisionamiento Astra Control". Para utilizar las funcionalidades del aprovisionador de control de Astra más recientes, necesitarás las versiones más recientes de Astra Trident y Astra Control Center.

- **Versión mínima de Astra Control Provisionador para usar con Astra Control Center**: Astra Control Provisionador 23,10 o posterior instalado y configurado.

Configuración de ONTAP con Astra Trident

- **Clase de almacenamiento**: Configure al menos una clase de almacenamiento en el clúster. Si se configura una clase de almacenamiento predeterminada, asegúrese de que es la única clase de almacenamiento con la designación predeterminada.
- **Controladores de almacenamiento y nodos de trabajo**: Asegúrese de configurar los nodos de trabajo en su clúster con los controladores de almacenamiento adecuados para que los pods puedan interactuar con el almacenamiento de backend. Astra Control Center es compatible con los siguientes controladores de ONTAP proporcionados por Astra Trident:
  - `ontap-nas`
  - `ontap-san`
  - `ontap-san-economy` (la replicación de aplicaciones no está disponible con este tipo de clase de almacenamiento)
  - `ontap-nas-economy` (las instantáneas y las políticas de replicación de aplicaciones no están disponibles con este tipo de clase de almacenamiento)

Back-ends de almacenamiento

Asegúrese de tener un backend soportado con capacidad suficiente.

- * Capacidad de almacenamiento de backend requerida*: Al menos 500GB disponibles
- **Backends soportados**: Astra Control Center soporta los siguientes backends de almacenamiento:

- Sistemas NetApp ONTAP 9.9.1 o posteriores AFF, FAS y ASA
- NetApp ONTAP Select 9.9.1 o posterior
- NetApp Cloud Volumes ONTAP 9.9.1 o posterior
- (Para la vista previa técnica de Centro de control de Astra) NetApp ONTAP 9.10.1 o posterior para operaciones de protección de datos que se proporcionan como versión preliminar técnica
- Longhorn 1.5.0 o posterior
  - Requiere la creación manual de un objeto VolumeSnapshotClass. Consulte la "Documentación de Longhorn" si desea obtener instrucciones.
- NetApp MetroCluster
  - Los clústeres de Kubernetes gestionados deben tener una configuración con ampliación.
- Back-ends de almacenamiento disponibles con proveedores de cloud admitidos

Licencias ONTAP

Para utilizar Astra Control Center, compruebe que dispone de las siguientes licencias de ONTAP, en función de lo que necesite:

- FlexClone
- SnapMirror: Opcional. Solo es necesario para la replicación en sistemas remotos mediante la tecnología SnapMirror. Consulte "Información sobre licencias de SnapMirror".
- Licencia de S3: Opcional. Solo se necesita para bloques ONTAP S3

Para comprobar si su sistema ONTAP tiene las licencias necesarias, consulte "Gestione licencias de ONTAP".

NetApp MetroCluster

Cuando usa NetApp MetroCluster como back-end de almacenamiento, tiene que hacer lo siguiente:

- Especifique una LIF de gestión de SVM como opción de back-end en el controlador de Astra Trident que utilice
- Asegúrese de tener la licencia de ONTAP adecuada

Para configurar el LIF MetroCluster, consulte estas opciones y ejemplos de cada controlador:

- "SAN"
- "NAS"

Licencia de Astra Control Center

Se requiere una licencia de Astra Control Center. Al instalar Astra Control Center, ya está activada una licencia de evaluación de 90 días para 4.800 CPU. Si necesita más capacidad o diferentes términos de evaluación, o si desea actualizar a una licencia completa, puede obtener otra licencia de evaluación o una licencia completa de NetApp. Necesita una licencia para proteger sus aplicaciones y datos.

Para probar Astra Control Center, regístrate para obtener una prueba gratuita. Puede registrarse registrándose "aquí".

Para configurar la licencia, consulte "utilice una licencia de evaluación de 90 días".

Para obtener más información sobre cómo funcionan las licencias, consulte "Licencia".

Requisitos de red

Configura tu entorno operativo para garantizar que Astra Control Center se pueda comunicar correctamente. Se requieren las siguientes configuraciones de red:

- **Dirección FQDN**: Debes tener una dirección FQDN para Astra Control Center.
- **Acceso a internet**: Debes determinar si tienes acceso externo a internet. Si no lo hace, es posible que algunas funcionalidades se vean limitadas, por ejemplo, enviar paquetes de soporte al "Sitio de soporte de NetApp".
- **Acceso al puerto**: El entorno operativo que aloja Astra Control Center se comunica mediante los siguientes puertos TCP. Debe asegurarse de que estos puertos estén permitidos a través de cualquier firewall y configurar firewalls para permitir que cualquier tráfico de salida HTTPS que se origine en la red Astra. Algunos puertos requieren conectividad de ambos modos entre el entorno que aloja Astra Control Center y cada clúster gestionado (se indica si procede).

Puede poner en marcha Astra Control Center en un clúster de Kubernetes de doble pila y Astra Control Center puede gestionar las aplicaciones y los back-ends de almacenamiento que se hayan configurado para un funcionamiento de doble pila. Para obtener más información sobre los requisitos de los clústeres de doble pila, consulte "Documentación de Kubernetes".

Origen	Destino	Puerto	Protocolo	Específico
PC cliente	Astra Control Center	443	HTTPS	Acceso IU/API: Asegúrese de que este puerto esté abierto en ambas direcciones entre Astra Control Center y el sistema utilizado para acceder a Astra Control Center
Consumidor de métricas	Nodo de trabajo de Astra Control Center	9090	HTTPS	Comunicación de datos de métricas: Asegúrese de que cada clúster gestionado pueda acceder a este puerto en el clúster que aloja a Astra Control Center (se requiere una comunicación bidireccional)
Astra Control Center	Proveedor de bloques de almacenamiento Amazon S3	443	HTTPS	Comunicación del almacenamiento de Amazon S3

Origen	Destino	Puerto	Protocolo	Específico
Astra Control Center	AutoSupport de NetApp	443	HTTPS	Comunicación AutoSupport de NetApp
Astra Control Center	Clúster de Kubernetes gestionado	443/6443 **NOTA**: El puerto que utiliza el clúster administrado puede variar dependiendo del clúster. Consulte la documentación del proveedor de software del clúster.	HTTPS	Comunicación con el clúster gestionado: Asegúrese de que este puerto esté abierto en ambos sentidos entre el clúster que aloja Astra Control Center y cada clúster gestionado

Entrada para clústeres de Kubernetes en las instalaciones

Puede elegir el tipo de entrada de red que utiliza Astra Control Center. De forma predeterminada, Astra Control Center implementa la puerta de enlace Astra Control Center (service/trafik) como un recurso para todo el clúster. Astra Control Center también admite el uso de un equilibrador de carga de servicio, si están permitidos en su entorno. Si prefiere utilizar un equilibrador de carga de servicio y aún no tiene uno configurado, puede utilizar el equilibrador de carga de MetalLB para asignar automáticamente 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.

> El equilibrador de carga debe utilizar una dirección IP ubicada en la misma subred que las direcciones IP del nodo de trabajo de Astra Control Center.

Para obtener más información, consulte "Configure la entrada para el equilibrio de carga".

Requisitos de incorporación de Google Anthos

Cuando alojes Astra Control Center en un clúster Anthos de Google, ten en cuenta que Google Anthos incluye de forma predeterminada el equilibrador de carga MetalLB y el servicio Istio Ingress, lo que te permite usar simplemente las capacidades genéricas de ingreso de Astra Control Center durante la instalación. Consulte "Documentación de instalación de Astra Control Center" para obtener más detalles.

Exploradores web compatibles

Astra Control Center es compatible con las versiones recientes de Firefox, Safari y Chrome con una resolución mínima de 1280 x 720.

Requisitos adicionales para clusters de aplicaciones

Tenga en cuenta estos requisitos si planea utilizar estas funciones de Astra Control Center:

- **Requisitos del clúster de aplicaciones**: "Requisitos de gestión de clústeres"
  - **Requisitos de aplicación gestionada**: "Y gestión de aplicaciones"
  - **Requisitos adicionales para la replicación de aplicaciones**: "Requisitos previos de replicación"

El futuro

Vea la "inicio rápido" descripción general.

Inicio rápido para Astra Control Center

A continuación se ofrece una descripción general de los pasos necesarios para empezar con Astra Control Center. Los vínculos de cada paso le llevan a una página que proporciona más detalles.

1 Revise los requisitos del clúster de Kubernetes

Asegúrese de que su entorno cumple estos requisitos:

Clúster de Kubernetes

- "Asegúrese de que el clúster de hosts cumple los requisitos de entorno operativo"
- "Configure el ingreso para el balanceo de carga en los clústeres de Kubernetes de las instalaciones"

Integración de almacenamiento

- "Compruebe que tu entorno incluye el aprovisionador de Astra Control"
- "Habilita las funciones avanzadas de gestión y aprovisionamiento de almacenamiento de Astra Control Provisioner"
- "Preparar nodos de trabajo de cluster"
- "Configurar los back-ends de almacenamiento"
- "Configure las clases de almacenamiento"
- "Instale una controladora Snapshot de volumen"
- "Cree una clase de snapshot de volumen"

Credenciales de ONTAP

- "Configure las credenciales de ONTAP"

2 Descargue e instale Astra Control Center

Complete estas tareas de instalación:

- "Descargue Astra Control Center desde la página de descargas del sitio de soporte de NetApp"
- Obtenga el archivo de licencia de NetApp:
  - Si está evaluando Astra Control Center, ya hay una licencia de evaluación integrada incluida
  - "Si ya ha adquirido Astra Control Center, genere su archivo de licencia"
- "Instalar Astra Control Center"
- "Realice pasos de configuración opcionales adicionales"

3 Complete algunas tareas de configuración inicial

Complete algunas tareas básicas para comenzar:

- "Añadir una licencia"
- "Preparar el entorno para la gestión de clústeres"
- "Añadir un clúster"
- "Añada un back-end de almacenamiento"
- "Añadir un bucket"

4 Utilice Astra Control Center

Cuando termine de configurar Astra Control Center, utiliza la interfaz de usuario de Astra Control o el "API de control Astra" para comenzar a administrar y proteger aplicaciones:

- "Gestionar cuentas": Usuarios, roles, LDAP, credenciales y más.
- "Gestionar notificaciones"
- "Gestionar aplicaciones": Definir recursos para gestionar.
- "Proteja sus aplicaciones": Configurar directivas de protección y replicar, clonar y migrar aplicaciones.

Si quiere más información

- "Utilice la API Astra Control"
- "Actualice Astra Control Center"
- "Obtenga ayuda con Astra Control"

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 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, el asesor digital 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-5fc9dc8d99-815zl               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:

```
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".

> 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 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 de 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.
- **Configurar el controlador de ingreso de Kubernetes**: Si tienes un controlador de ingreso de Kubernetes que administre el acceso externo a los servicios, como el balanceo de carga en un clúster, debes

configurarlo para usarlo 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) 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 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
```

Pasos

- Descargue y extraiga Astra Control Center
- Complete los pasos adicionales si utiliza un registro local
- Busque la página de instalación del operador
- Instale el operador
- Instalar 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. 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 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
```

Busque la página de instalación del operador

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

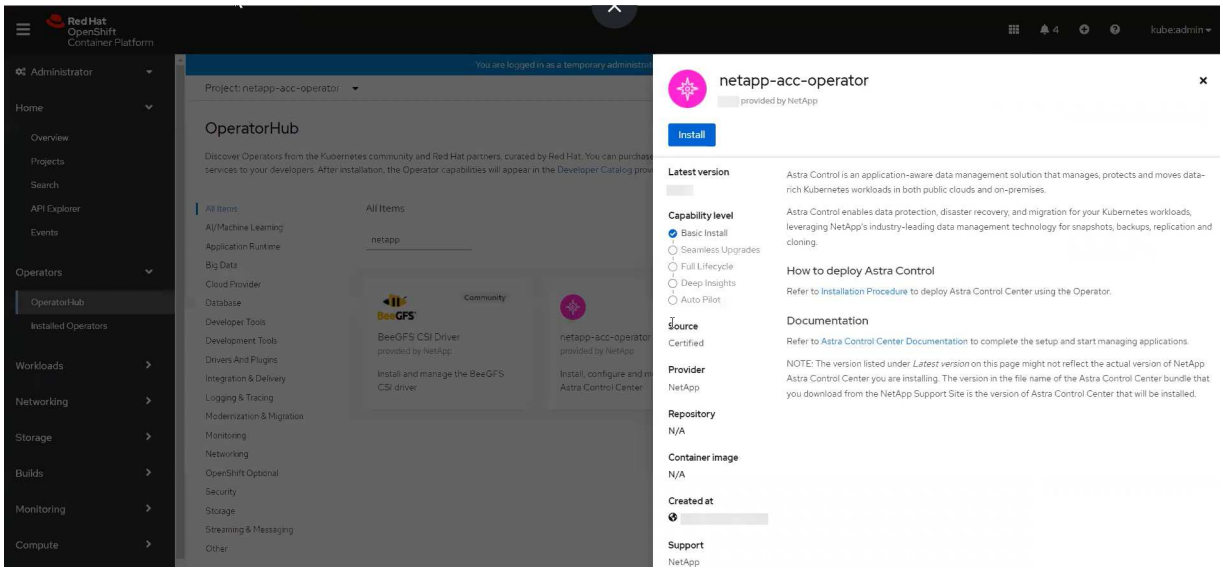
Consola web de Red Hat OpenShift

a. Inicie sesión en la IU de OpenShift Container Platform.

b. En el menú lateral, seleccione **operadores > OperatorHub**.

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

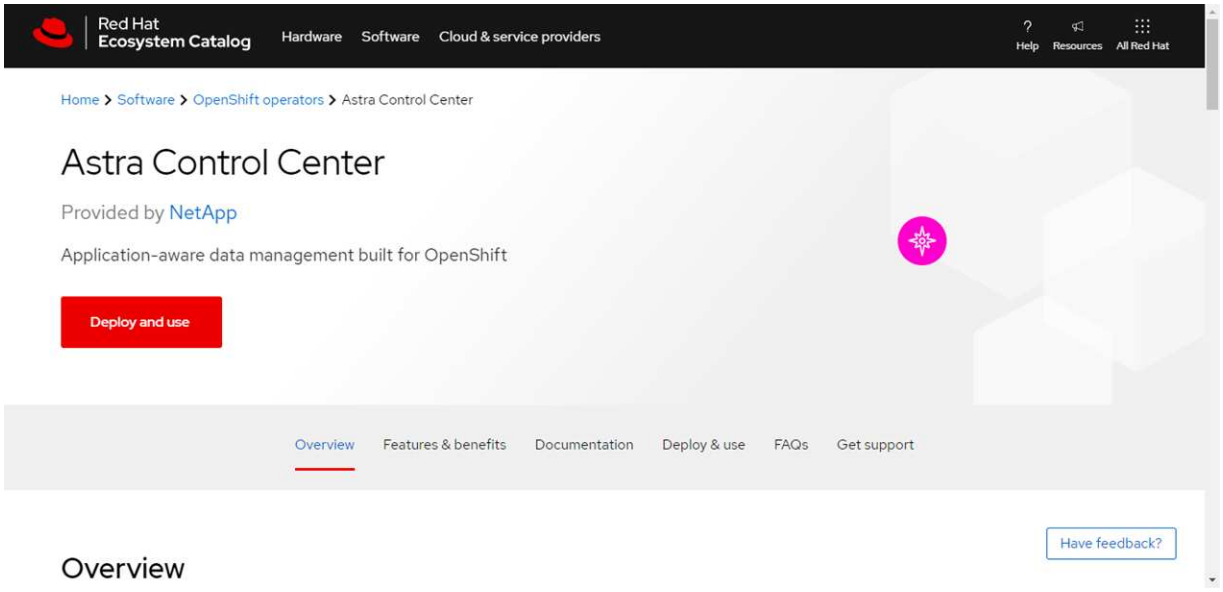
c. Busque `netapp-acc` Y selecciona el operador Centro de control de Astra de NetApp.



Catálogo de Red Hat Ecosystem

a. Seleccione Astra Control Center de NetApp "operador".

b. Seleccione **Desplegar y usar**.



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**.

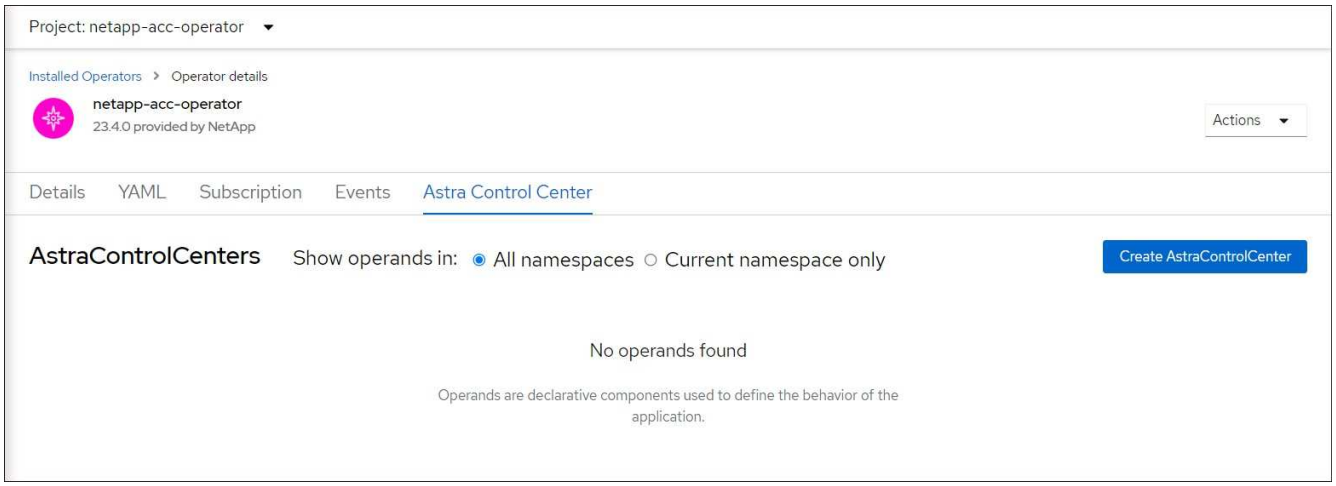
      Si ha seleccionado una estrategia de aprobación manual, se le pedirá que apruebe el plan de instalación manual para este operador.

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, 24.02.0-69.

   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 tamaño de escala 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:

- **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.

- **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 **Image Registry**, utilice el valor predeterminado a menos que configure un registro local. Para un registro local, reemplace este valor por la ruta del registro de imágenes local donde insertó las imágenes en un paso anterior. 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, deberá realizar el trabajo "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. Puedes poner en marcha Astra Control Center en tus clústeres de Kubernetes on-premises o en uno de los clústeres de Kubernetes autogestionados del entorno de nube.

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 implementar Astra Control Center en AWS, necesitarás 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 recursos específicos además de los requisitos de recursos del entorno. Consulte "Requisitos del entorno operativo del Centro de control de Astra".

> El token de registro de AWS caduca en 12 horas, después de lo cual tendrá que renovar el secreto del registro de imágenes de 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 al registro de imágenes del Centro de control de Astra de NetApp, también deberá crear un registro de contenedores elásticos (ECR) para alojar las imágenes del Centro de control de Astra e insertar las imágenes en este registro.

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

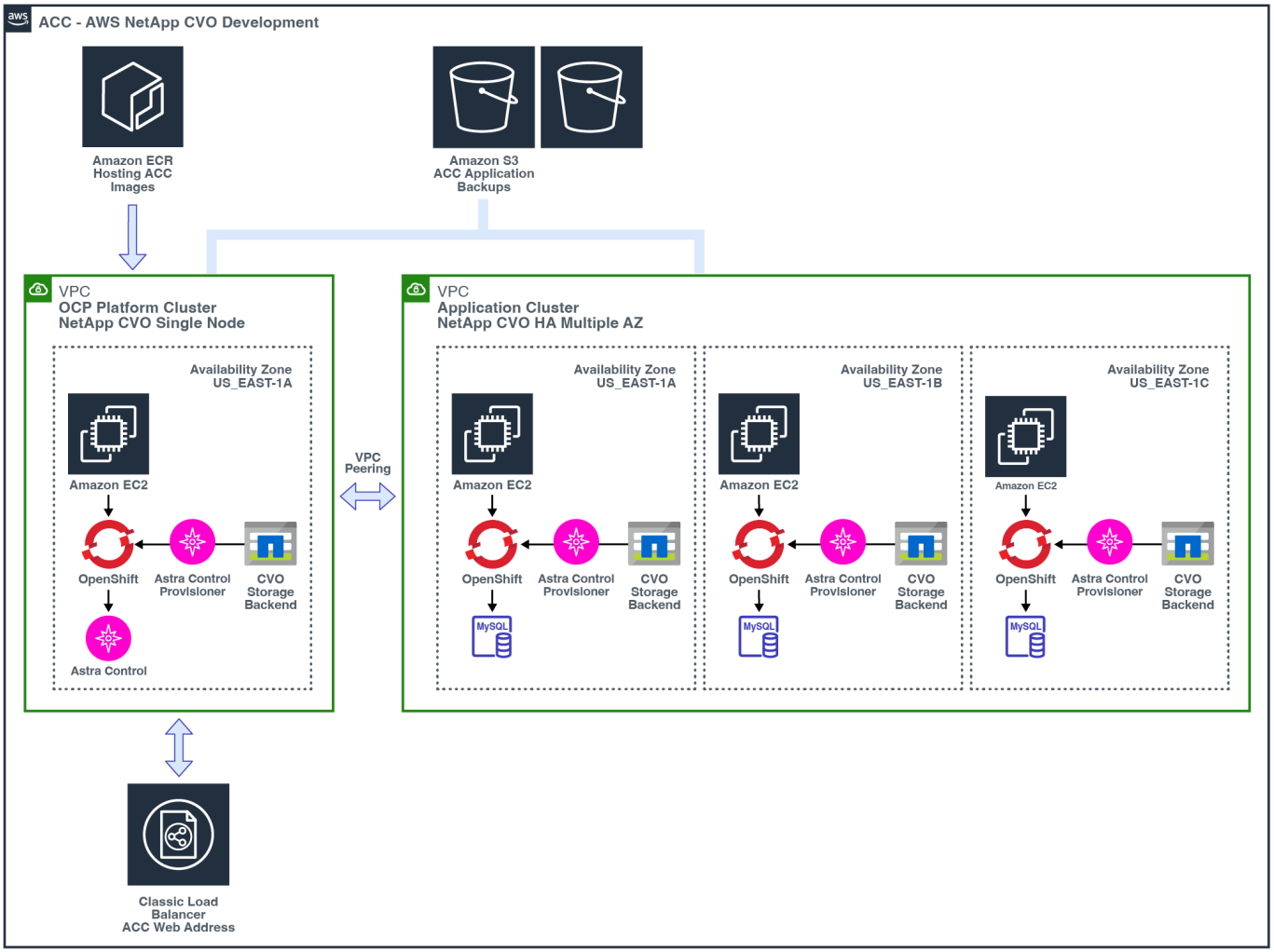
1. Cree una red virtual AWS.
2. Revise las instancias de computación EC2. Puede ser un servidor con configuración básica o máquinas virtuales en AWS.
3. Si el tipo de instancia no coincide con los requisitos mínimos de recursos de Astra para los nodos maestros y trabajadores, cambie el tipo de instancia 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 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 aprovisionamiento de Astra Control.

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 Control Provisioning 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, necesitarás 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 recursos específicos además de los requisitos de recursos del entorno. Consulte "Requisitos del entorno operativo del Centro de control de Astra".

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. Instale 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. Configure GCP.
5. Configuración de NetApp BlueXP para GCP.
6. Instale Astra Control Center para GCP.

Instale 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".

Configure GCP

A continuación, configure GCP para crear una VPC, configurar instancias de computación y crear un almacenamiento de objetos de Google Cloud. Si no puedes acceder al registro de imágenes del Centro de control de Astra de NetApp, también tendrás que crear un Registro de contenedores de Google para alojar las imágenes del Centro de control de Astra e insertar las imágenes en 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 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

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

Siga la documentación de BlueXP para completar los siguientes pasos. Consulte "Introducción a 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. Agregue 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, observa la versión de aprovisionamiento de Astra Control.
   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 Control Provisioning 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.

Instale Astra Control Center para GCP

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

> GCP utiliza el tipo de bloque 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 implementar Astra Control Center en Azure, necesitarás 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 recursos específicos además de los requisitos de recursos del entorno. Consulte "Requisitos del entorno operativo del Centro de control de Astra".

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 al registro de imágenes del Centro de control de Astra de NetApp, también deberá crear un Registro de contenedores de Azure (ACR) para alojar las imágenes del Centro de control de Astra e insertar las imágenes en 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. Necesitarás una cuenta de almacenamiento para crear un contenedor que se utilice 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 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".

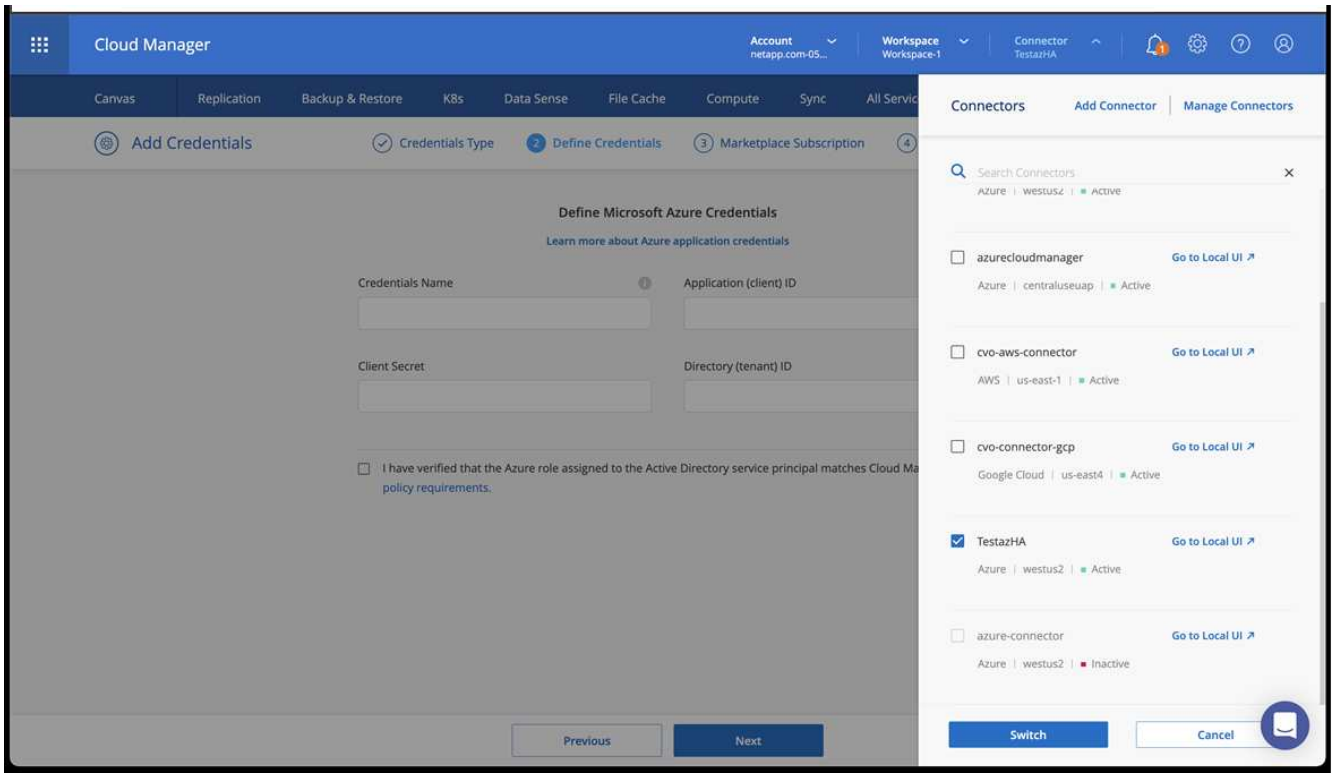
Antes de empezar

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

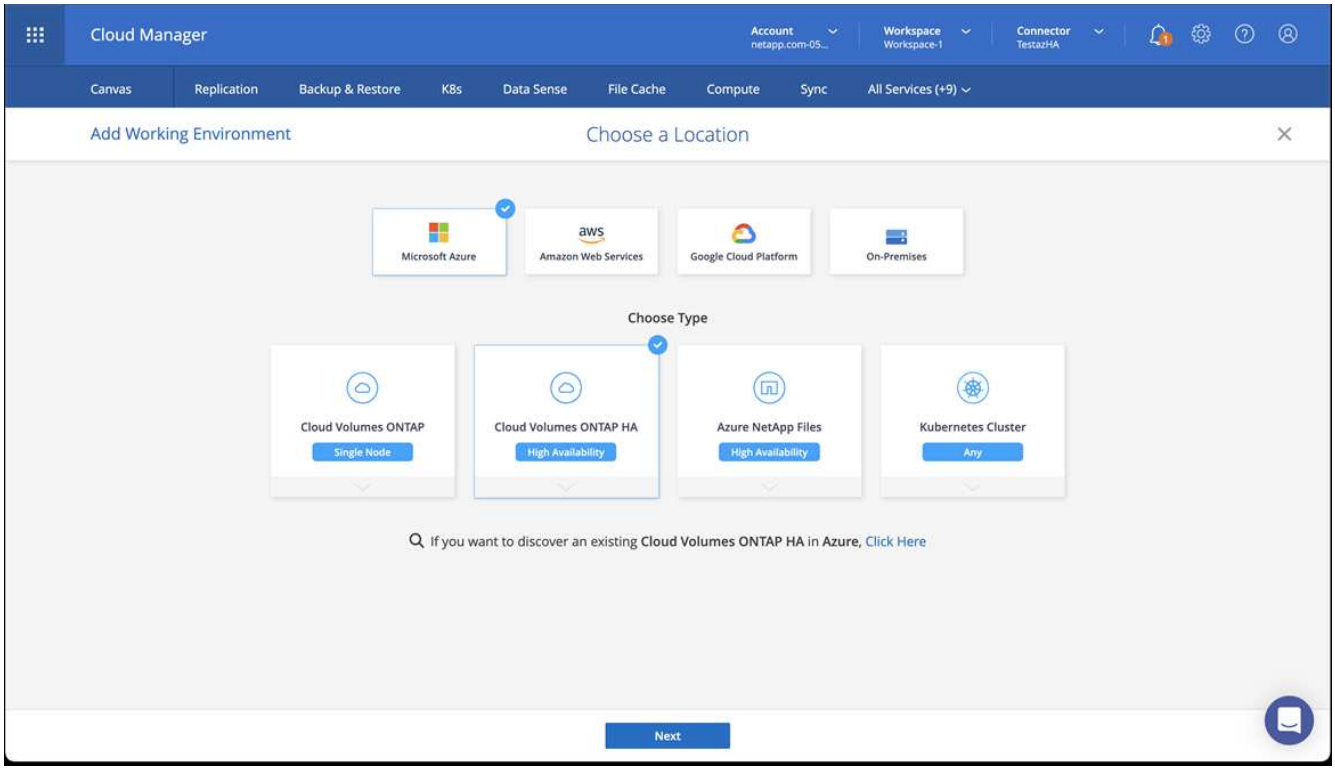
Pasos

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.

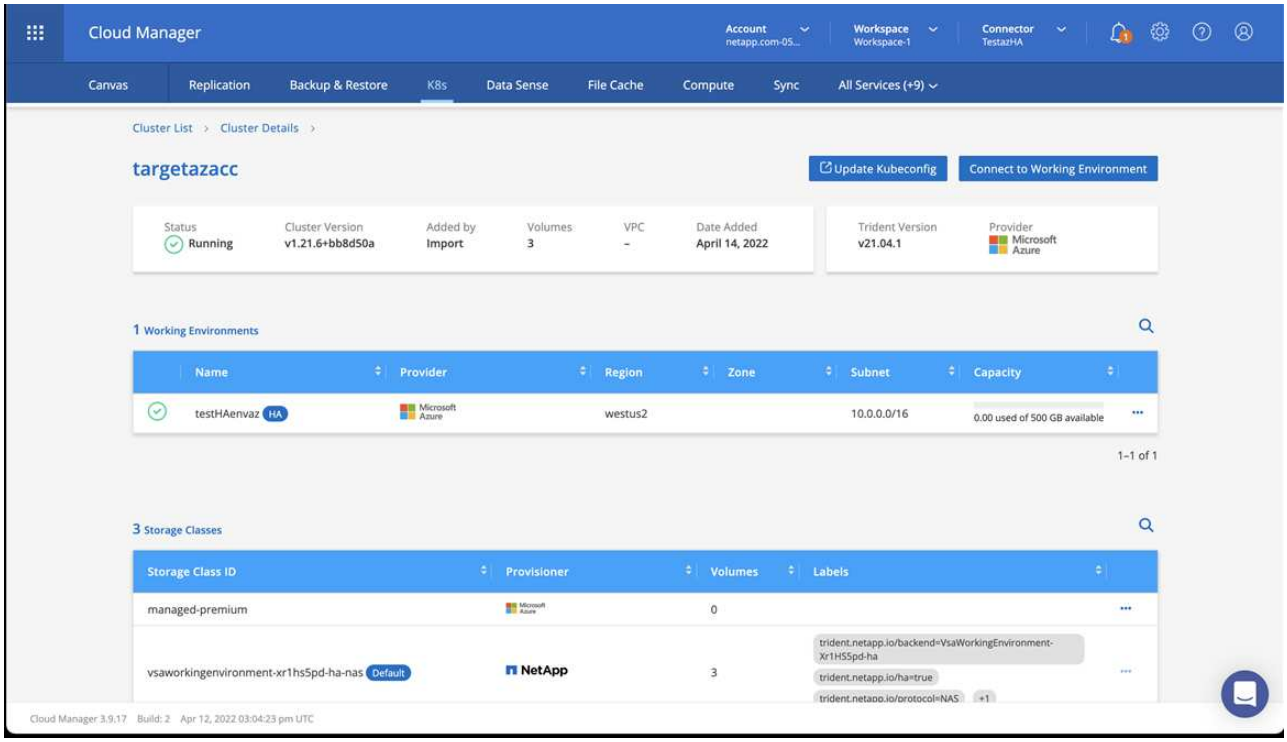


4. Cree un entorno de trabajo para su entorno de cloud.
   a. Ubicación: "Microsoft Azure".
   b. Tipo: "Cloud Volumes ONTAP ha".



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 aprovisionamiento de Astra Control.

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 Control Provisioning 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. Para el uso en producción, debe quitar el certificado TLS autofirmado existente y reemplazarlo por 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:

> En este ejemplo se utiliza el `dnsNames` Propiedad para especificar la dirección DNS de Astra Control Center. Astra Control Center no admite el uso de la propiedad Common Name (CN) para especificar la dirección DNS.

```
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.

Configure Astra Control Center

Agregue una licencia de Astra Control Center

Al instalar Astra Control Center, ya hay una licencia de evaluación integrada instalada. Si estás evaluando Astra Control Center, puedes omitir este paso.

Puede añadir una nueva licencia con la interfaz de usuario de Astra Control o. "API de control Astra".

Las licencias de Astra Control Center miden los recursos de CPU mediante unidades de CPU de Kubernetes y representan los recursos de CPU asignados a los nodos de trabajo de todos los clústeres de Kubernetes gestionados. Las licencias se basan en el uso de vCPU. Para obtener más información sobre cómo se calculan las licencias, consulte "Licencia".

> Si su instalación crece para superar el número de unidades CPU con licencia, Astra Control Center le impide gestionar nuevas aplicaciones. Se muestra una alerta cuando se supera la capacidad.

> Para actualizar una evaluación existente o una licencia completa, consulte "Actualizar una licencia existente".

Antes de empezar

- Acceso a una instancia de Astra Control Center recién instalada.
- Permisos del rol de administrador.
- A. "Archivo de licencia de NetApp" (NLF).

Pasos

1. Inicie sesión en la interfaz de usuario de Astra Control Center.
2. Seleccione **cuenta** > **Licencia**.
3. Seleccione **Agregar licencia**.
4. Busque el archivo de licencia (NLF) que descargó.
5. Seleccione **Agregar licencia**.

La página **cuenta** > **Licencia** muestra la información de la licencia, la fecha de caducidad, el número de serie de la licencia, el ID de cuenta y las unidades de CPU utilizadas.

Si tiene una licencia de evaluación y no envía datos a AutoSupport, asegúrese de almacenar su ID de cuenta para evitar la pérdida de datos en caso de un fallo en Astra Control Center.

Habilita el aprovisionador de Astra Control

Las versiones 23,10 y posteriores de Astra Trident incluyen la opción de usar Astra Control Provisioning, que permite a los usuarios de Astra Control con licencia acceder a funcionalidades avanzadas de aprovisionamiento del almacenamiento. El aprovisionador Astra Control ofrece esta funcionalidad ampliada, además de la funcionalidad estándar basada en CSI de Astra Trident.

En las próximas actualizaciones de Astra Control, el aprovisionador de Astra Control reemplazará a Astra Trident como aprovisionador de almacenamiento y orquestador y será obligatorio para su uso en Astra Control. Por este motivo, se recomienda encarecidamente que los usuarios de Astra Control habiliten el aprovisionador de Astra Control. Astra Trident seguirá siendo de código abierto y se seguirá lanzando, manteniendo, admitiendo y actualizando con las nuevas funciones CSI y otras de NetApp.

Acerca de esta tarea

Debes seguir este procedimiento si eres un usuario del Centro de control de Astra con licencia y quieres utilizar la funcionalidad de aprovisionamiento de Astra Control. También debes seguir este procedimiento si eres usuario de Astra Trident y quieres utilizar la funcionalidad adicional que proporciona el aprovisionador de Astra Control sin utilizar también Astra Control.

En cada caso, la funcionalidad de aprovisionador no está habilitada de manera predeterminada en Astra Trident 24,02 y debe estar habilitada.

Antes de empezar

Si habilita el aprovisionador de Astra Control, primero haga lo siguiente:

Astra Control proporciona a los usuarios aprovisionamiento con Astra Control Center

- **Obtén una licencia de Astra Control Center**: Necesitarás una "Licencia de Astra Control Center" Para habilitar el aprovisionador de Astra Control y acceder a las funcionalidades que ofrece.
- **Instalar o actualizar a Astra Control Center 23,10 o posterior**: Necesitarás la última versión de Astra Control Center (24,02) si planeas usar la última funcionalidad de Astra Control Provisionador (24,02) con Astra Control.
- **Confirme que su clúster tiene una arquitectura de sistema AMD64**: La imagen del aprovisionador de Astra Control se proporciona en las arquitecturas de CPU AMD64 y ARM64, pero solo AMD64 es compatible con Astra Control Center.
- **Obtén una cuenta del Servicio de control de Astra para acceder al registro**: Si tienes la intención de usar el Registro de control de Astra en lugar del Sitio de soporte de NetApp para descargar la imagen del aprovisionador de control de Astra, completa el registro para un "Cuenta de Astra Control Service". Después de completar, enviar el formulario y crear una cuenta de BlueXP, recibirás un correo electrónico de bienvenida de Astra Control Service.
- **Si tienes Astra Trident instalado, confirma que su versión está dentro de una ventana de cuatro versiones**: Puedes realizar una actualización directa a Astra Trident 24,02 con el aprovisionador de control de Astra si tu Astra Trident está dentro de una ventana de cuatro versiones de la versión 24,02. Por ejemplo, puedes actualizar directamente de Astra Trident 23,04 a 24,02.

El aprovisionador de Astra Control solo para los usuarios

- **Obtén una licencia de Astra Control Center**: Necesitarás una "Licencia de Astra Control Center" Para habilitar el aprovisionador de Astra Control y acceder a las funcionalidades que ofrece.
- **Si tienes Astra Trident instalado, confirma que su versión está dentro de una ventana de cuatro versiones**: Puedes realizar una actualización directa a Astra Trident 24,02 con el aprovisionador de control de Astra si tu Astra Trident está dentro de una ventana de cuatro versiones de la versión 24,02. Por ejemplo, puedes actualizar directamente de Astra Trident 23,04 a 24,02.
- **Obtén una cuenta de Astra Control Service para acceder al registro**: Necesitarás acceder al registro para descargar imágenes de Astra Control Provisionador. Para comenzar, complete el registro para una "Cuenta de Astra Control Service". Después de completar, enviar el formulario y crear una cuenta de BlueXP, recibirás un correo electrónico de bienvenida de Astra Control Service.

(Paso 1) Obtén la imagen del aprovisionador de Astra Control

Los usuarios de Astra Control Center pueden obtener la imagen del aprovisionador de control de Astra mediante el registro de Astra Control o el método del sitio de soporte de NetApp. Los usuarios de Astra Trident que deseen utilizar el aprovisionador de control de Astra sin Astra Control deben utilizar el método de registro.

Registro de imágenes de Astra Control

Puede utilizar Podman en lugar de Docker para los comandos de este procedimiento. Si se utiliza un entorno de Windows, se recomienda PowerShell.

1. Acceda al registro de imágenes de Astra Control de NetApp:
   a. Inicie sesión en la interfaz de usuario web de Astra Control Service y seleccione el icono de figura situado en la parte superior derecha de la página.
   b. Seleccione **acceso API**.
   c. Escriba su ID de cuenta.
   d. 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.
   e. Inicia sesión en el registro de Astra Control usando el método que prefieras:

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

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

2. (Solo registros personalizados) Siga estos pasos para mover la imagen a su registro personalizado. Si no está utilizando un registro, siga los pasos del operador Trident en la "siguiente sección".
   a. Extrae la imagen del aprovisionador de Astra Control del registro:

   La imagen extraída no soportará múltiples plataformas y solo soportará la misma plataforma que el host que sacó la imagen, como Linux AMD64.

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0 --platform <cluster platform>
```

   Ejemplo:

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0 --platform linux/amd64
```

   a. Etiquete la imagen:

```
docker tag cr.astra.netapp.io/astra/trident-acp:24.02.0 <my_custom_registry>/trident-acp:24.02.0
```

   b. Introduzca la imagen en el registro personalizado:

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

Puede utilizar Crane copy como alternativa a la ejecución de estos comandos Docker: `crane copy cr.astra.netapp.io/astra/trident-acp:24.02.0 <my_custom_registry>/trident-acp:24.02.0`

Sitio de soporte de NetApp

1. Descarga el bundle Astra Control Provisioner (`trident-acp-[version].tar`) 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 trident-acp-[version].

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

```
openssl dgst -sha256 -verify certs/AstraControlCenterDockerImages-public.pub -signature certs/trident-acp-[version].tar.sig trident-acp-[version].tar
```

3. Cargue la imagen del aprovisionador de Astra Control:

```
docker load < trident-acp-24.02.0.tar
```

Respuesta:

```
Loaded image: trident-acp:24.02.0-linux-amd64
```

4. Etiquete la imagen:

```
docker tag trident-acp:24.02.0-linux-amd64 <my_custom_registry>/trident-acp:24.02.0
```

5. Introduzca la imagen en el registro personalizado:

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

(Paso 2) Habilitar el aprovisionador de Astra Control en Astra Trident

Determine si el método de instalación original ha utilizado un "Operador (manualmente o con Helm) o tridentctl" y complete los pasos apropiados de acuerdo con su método original.

Operador Astra Trident

1. "Descarga el instalador de Astra Trident y extráigalo".
2. Complete estos pasos si todavía no ha instalado Astra Trident o si ha quitado el operador de la implementación original de Astra Trident:
   a. Cree el CRD:

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.y
aml
```

   b. Cree el espacio de nombres trident (`kubectl create namespace trident`) o confirme que el espacio de nombres trident sigue existiendo (`kubectl get all -n trident`). Si el espacio de nombres se ha eliminado, vuelva a crearlo.
3. Actualice Astra Trident a 24.02.0:

> Para los clústeres que ejecutan Kubernetes 1,24 o una versión anterior, utilice `bundle_pre_1_25.yaml`. Para los clústeres que ejecutan Kubernetes 1,25 o posterior, utilice `bundle_post_1_25.yaml`.

```
kubectl -n trident apply -f trident-installer/deploy/<bundle-
name.yaml>
```

4. Compruebe que Astra Trident está ejecutando:

```
kubectl get torc -n trident
```

   Respuesta:

```
NAME      AGE
trident   21m
```

5. Si tienes un registro que usa secretos, crea un secreto para extraer la imagen del aprovisionador de Astra Control:

```
kubectl create secret docker-registry <secret_name> -n trident
--docker-server=<my_custom_registry> --docker-username=<username>
--docker-password=<token>
```

6. Edite el CR de TridentOrchestrator y realice las siguientes modificaciones:

```
kubectl edit torc trident -n trident
```

a. Establezca una ubicación de registro personalizada para la imagen de Astra Trident o extráigala del registro de Astra Control (`tridentImage: <my_custom_registry>/trident:24.02.0` o. `tridentImage: netapp/trident:24.02.0`).

b. Habilita el aprovisionador de Astra Control (`enableACP: true`).

c. Establezca la ubicación de registro personalizada para la imagen del aprovisionador de Astra Control o sáquela del registro de Astra Control (`acpImage: <my_custom_registry>/trident-acp:24.02.0` o. `acpImage: cr.astra.netapp.io/astra/trident-acp:24.02.0`).

d. Si estableció la imagen descubre los secretos anteriormente en este procedimiento, puede establecerlos aquí (`imagePullSecrets: - <secret_name>`). Utilice el mismo nombre secreto que estableció en los pasos anteriores.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  tridentImage: <registry>/trident:24.02.0
  enableACP: true
  acpImage: <registry>/trident-acp:24.02.0
  imagePullSecrets:
  - <secret_name>
```

7. Guarde y salga del archivo. El proceso de despliegue comenzará automáticamente.
8. Compruebe que se han creado el operador, el despliegue y los replicasets.

```
kubectl get all -n trident
```

Solo debe haber **una instancia** del operador en un clúster de Kubernetes. No cree varias implementaciones del operador Trident de Astra.

9. Compruebe el `trident-acp` container se está ejecutando y eso `acpVersion` es `24.02.0` con el estado de `Installed`:

```
kubectl get torc -o yaml
```

Respuesta:

```
status:
  acpVersion: 24.02.0
  currentInstallationParams:
    ...
    acpImage: <registry>/trident-acp:24.02.0
    enableACP: "true"
    ...
  ...
  status: Installed
```

tridentctl

1. "Descarga el instalador de Astra Trident y extráigalo".
2. "Si ya tiene un Astra Trident existente, desinstálelo del clúster que lo aloja".
3. Instale Astra Trident con el aprovisionador de control de Astra habilitado (`--enable-acp=true`):

```
./tridentctl -n trident install --enable-acp=true --acp
-image=mycustomregistry/trident-acp:24.02
```

4. Confirme que se ha habilitado el aprovisionador de Astra Control:

```
./tridentctl -n trident version
```

Respuesta:

```
+----------------+----------------+-------------+ | SERVER VERSION |
CLIENT VERSION | ACP VERSION | +----------------+----------------
+-------------+ | 24.02.0 | 24.02.0 | 24.02.0. | +----------------
+----------------+-------------+
```

Timón

1. Si tiene Astra Trident 23.07.1 o anterior instalado, "desinstalar" el operador y otros componentes.
2. Si tu clúster de Kubernetes ejecuta la versión 1,24 o anterior, elimina psp:

```
kubectl delete psp tridentoperatorpod
```

3. Añada el repositorio de Astra Trident Helm:

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-
chart
```

4. Actualice el gráfico Helm:

```
helm repo update netapp-trident
```

Respuesta:

```
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "netapp-trident" chart
repository
Update Complete. □Happy Helming!□
```

5. Enumere las imágenes:

```
./tridentctl images -n trident
```

Respuesta:

```
| v1.28.0            | netapp/trident:24.02.0|
|                    | docker.io/netapp/trident-autosupport:24.02|
|                    | registry.k8s.io/sig-storage/csi-
provisioner:v4.0.0|
|                    | registry.k8s.io/sig-storage/csi-
attacher:v4.5.0|
|                    | registry.k8s.io/sig-storage/csi-
resizer:v1.9.3|
|                    | registry.k8s.io/sig-storage/csi-
snapshotter:v6.3.3|
|                    | registry.k8s.io/sig-storage/csi-node-driver-
registrar:v2.10.0 |
|                    | netapp/trident-operator:24.02.0 (optional)
```

6. Asegúrese de que el trident-operator 24.02.0 esté disponible:

```
helm search repo netapp-trident/trident-operator --versions
```

Respuesta:

```
NAME                              CHART VERSION   APP VERSION
DESCRIPTION
netapp-trident/trident-operator 100.2402.0      24.02.0         A
```

7. Uso `helm install` y ejecute una de las siguientes opciones que incluyen estos ajustes:
   - Un nombre para la ubicación de despliegue
   - La versión de Trident de Astra
   - El nombre de la imagen del aprovisionador de Astra Control
   - La marca para habilitar el aprovisionador
   - (Opcional) Una ruta de registro local. Si está utilizando un registro local, su "Imágenes de Trident" Se pueden ubicar en un registro o en diferentes registros, pero todas las imágenes CSI deben estar ubicadas en el mismo registro.
   - El espacio de nombres de Trident

Opciones

- Imágenes sin registro

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=cr.astra.netapp.io/astra/trident-acp:24.02.0
--set enableACP=true --set operatorImage=netapp/trident-
operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

- Imágenes en uno o más registros

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=<your-registry>:<acp image> --set
enableACP=true --set imageRegistry=<your-registry>/sig-storage --set
operatorImage=netapp/trident-operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

Puede utilizar `helm list` para revisar detalles de la instalación como nombre, espacio de nombres, gráfico, estado, versión de la aplicación, y el número de revisión.

Si tiene problemas para poner en marcha Trident mediante Helm, ejecute este comando para desinstalar completamente Astra Trident:

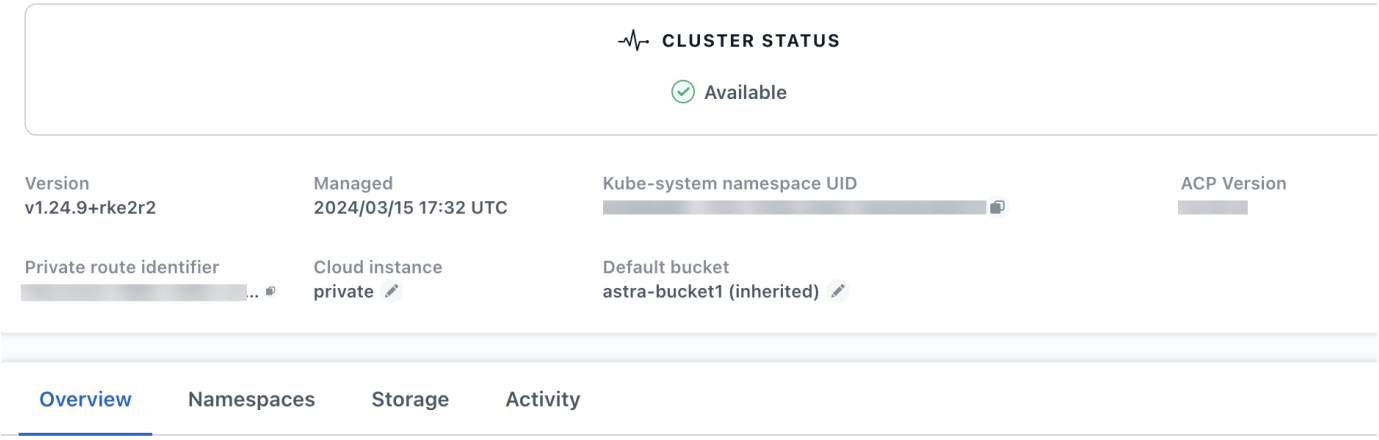
```
./tridentctl uninstall -n trident
```

No "Elimina por completo los CRD de Astra Trident" Como parte de la desinstalación antes de intentar habilitar de nuevo Astra Control Provisioner.

Resultado

Está habilitada la funcionalidad de aprovisionamiento de Astra Control y es posible usar cualquier función disponible para la versión que esté ejecutando.

(Solo para usuarios de Astra Control Center) Después de instalar Astra Control Provisioner, el clúster que aloja el aprovisionador en la interfaz de usuario de Astra Control Center mostrará un `ACP version` en lugar de `Trident version` campo y núm. de versión instalada actual.



Si quiere más información

- "Documentación sobre actualizaciones de Astra Trident"

Prepare su entorno para la gestión de clústeres con Astra Control

Antes de añadir un clúster, debe asegurarse de que se cumplen las siguientes condiciones previas. También debe realizar comprobaciones de elegibilidad para asegurarse de que su clúster esté listo para añadirse a Astra Control Center y crear roles de clúster kubeconfig según sea necesario.

Astra Control le permite añadir clústeres gestionados mediante recurso personalizado (CR) o kubeconfig, en función de su entorno y sus preferencias.

Antes de empezar

- **Cumplir con los requisitos ambientales**: Su entorno cumple "requisitos del entorno operativo" Para Astra Control Center.
- * Configurar nodos de trabajador*: Asegúrese de que usted "configure los nodos de trabajo" en su clúster con los controladores de almacenamiento adecuados para que los pods puedan interactuar con el almacenamiento back-end.
- **Habilitar restricciones PSA**: Si su clúster tiene activada la aplicación de admisión de seguridad de pod, que es estándar para los clústeres de Kubernetes 1,25 y posteriores, debe habilitar las restricciones PSA en estos espacios de nombres:
  - `netapp-acc-operator` espacio de nombres:

```
kubectl label --overwrite ns netapp-acc-operator pod-
security.kubernetes.io/enforce=privileged
```

- `netapp monitoring` espacio de nombres:

```
kubectl label --overwrite ns netapp-monitoring pod-
security.kubernetes.io/enforce=privileged
```

- **Credenciales de ONTAP**: Necesita credenciales de ONTAP y un superusuario e ID de usuario establecidos en el sistema ONTAP de respaldo para realizar copias de seguridad y restaurar aplicaciones con Astra Control Center.

  Ejecute los siguientes comandos en la línea de comandos de la ONTAP:

```
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -superuser sys
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -anon 65534
```

- **Requisitos de clúster gestionados por kubeconfig**: Estos requisitos son específicos para los clusters de aplicaciones gestionados por kubeconfig.
  - **Hacer kubeconfig accesible**: Usted tiene acceso a la "kubeconfig de cluster por defecto" eso "ha configurado durante la instalación".
  - **Consideraciones de la autoridad de certificación**: Si está agregando el clúster usando un archivo kubeconfig que hace referencia a una autoridad de certificación (CA) privada, agregue la siguiente línea a la `cluster` sección del archivo kubeconfig. Esto permite a Astra Control añadir el clúster:

```
insecure-skip-tls-verify: true
```

  - **Sólo rancher**: Al administrar clústeres de aplicaciones en un entorno Rancher, modifique el contexto predeterminado del clúster de aplicaciones en el archivo kubeconfig proporcionado por Rancher para utilizar un contexto de plano de control en lugar del contexto del servidor API Rancher. Esto reduce la carga en el servidor API de Rancher y mejora el rendimiento.
- **Requisitos del aprovisionador de Astra Control**: Debes tener un aprovisionador de Astra Control configurado correctamente, incluidos sus componentes de Astra Trident, para gestionar clústeres.
  - **Revise los requisitos del entorno de Astra Trident**: Antes de instalar o actualizar el aprovisionador de Astra Control, revise el "compatibles con front-ends, back-ends y configuraciones de host".
  - **Habilitar la funcionalidad de aprovisionamiento de Astra Control**: Se recomienda instalar Astra Trident 23,10 o posterior y activar "Funcionalidad de almacenamiento avanzada de Astra Control Provisioning". En las siguientes versiones, Astra Control no será compatible con Astra Trident si el aprovisionador de Astra Control también no está habilitado.
  - **Configurar un backend de almacenamiento**: Al menos un backend de almacenamiento debe ser "Configuradas en Astra Trident" en el clúster.

- **Configure una clase de almacenamiento**: Al menos una clase de almacenamiento debe ser "Configuradas en Astra Trident" en el clúster. Si se configura una clase de almacenamiento predeterminada, asegúrese de que es la clase de almacenamiento **Only** que tiene la anotación predeterminada.
- **Configure un controlador de instantáneas de volumen e instale una clase de instantáneas de volumen**: "Instale una controladora Snapshot de volumen" Para poder crear instantáneas en Astra Control. "Cree" al menos uno `VolumeSnapshotClass` Mediante Astra Trident.

Ejecutar las comprobaciones de elegibilidad

Ejecute las siguientes comprobaciones de elegibilidad para asegurarse de que su clúster esté listo para ser agregado a Astra Control Center.

Pasos

1. Determine la versión de Astra Trident que ejecuta:

   ```
   kubectl get tridentversion -n trident
   ```

   Si existe Astra Trident, obtendrá un resultado similar al siguiente:

   ```
   NAME      VERSION
   trident   24.02.0
   ```

   Si Astra Trident no existe, obtendrá un resultado similar al siguiente:

   ```
   error: the server doesn't have a resource type "tridentversions"
   ```

2. Debe realizar una de las siguientes acciones:
   - Si utiliza Astra Trident 23,01 o una versión anterior, utilice estos "instrucciones" Para actualizar a una versión más reciente de Astra Trident antes de actualizar a Astra Control Provisioner. Puede hacerlo "realice una actualización directa" Para Astra Control Provisioner 24,02 si tu Astra Trident está en una ventana de cuatro versiones de la versión 24,02. Por ejemplo, puedes actualizar directamente de Astra Trident 23,04 a Astra Control Provisioner 24,02.
   - Si utiliza Astra Trident 23,10 o una versión posterior, compruebe que el aprovisionador de Astra Control haya sido "activado". El aprovisionador de Astra Control no funcionará con versiones de Astra Control Center anteriores a la 23,10. "Actualiza tu aprovisionador de Astra Control" De modo que tiene la misma versión que Astra Control Center que vas a actualizar para acceder a la funcionalidad más reciente.
3. Asegúrese de que todos los pods (incluidos `trident-acp`) se están ejecutando:

   ```
   kubectl get pods -n trident
   ```

4. Determine si las clases de almacenamiento están utilizando los controladores Astra Trident compatibles. El nombre del aprovisionador debe ser `csi.trident.netapp.io`. Consulte el siguiente ejemplo:

```
kubectl get sc
```

Respuesta de ejemplo:

```
NAME                   PROVISIONER            RECLAIMPOLICY
VOLUMEBINDINGMODE  ALLOWVOLUMEEXPANSION  AGE
ontap-gold (default)  csi.trident.netapp.io  Delete         Immediate
true                   5d23h
```

Crear un rol de cluster kubeconfig

En el caso de los clústeres que se gestionan mediante kubeconfig, puede crear una función de administrador de permisos limitados o de permisos ampliados para Astra Control Center. Este no es un procedimiento obligatorio para la configuración de Astra Control Center, ya que ya configuró un kubeconfig como parte de la "proceso de instalación".

Este procedimiento le ayuda a crear un kubeconfig independiente si cualquiera de los siguientes escenarios se aplica a su entorno:

- Deseas limitar los permisos de Astra Control a los clústeres que gestiona
- Usas varios contextos y no puedes usar el comando predeterminado de Astra Control configurado durante la instalación o un rol limitado con un solo contexto no funcionará en tu entorno

Antes de empezar

Asegúrese de que tiene lo siguiente para el clúster que tiene intención de administrar antes de completar los pasos del procedimiento:

- kubectl v1.23 o posterior instalado
- Acceda con atención al clúster que pretende añadir y gestionar con Astra Control Center

> Para este procedimiento, no necesita acceso kubectl al clúster que ejecuta Astra Control Center.

- Una imagen de referencia activa para el clúster que pretende gestionar con derechos de administrador del clúster para el contexto activo

Pasos

1. Cree una cuenta de servicio:
   a. Cree un archivo de cuenta de servicio llamado `astracontrol-service-account.yaml`.

```
<strong>astracontrol-service-account.yaml</strong>
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astracontrol-service-account
  namespace: default
```

b. Aplicar la cuenta de servicio:

```
kubectl apply -f astracontrol-service-account.yaml
```

2. Cree uno de los siguientes roles de clúster con permisos suficientes para que Astra Control gestione un clúster:

Rol de clúster limitado

Este rol contiene los permisos mínimos necesarios para que Astra Control gestione un clúster:

a. Cree un `ClusterRole` archivo llamado, por ejemplo, `astra-admin-account.yaml`.

```
<strong>astra-admin-account.yaml</strong>
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:

# Get, List, Create, and Update all resources
# Necessary to backup and restore all resources in an app
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - get
  - list
  - create
  - patch

# Delete Resources
# Necessary for in-place restore and AppMirror failover
- apiGroups:
  - ""
  - apps
  - autoscaling
  - batch
  - crd.projectcalico.org
  - extensions
  - networking.k8s.io
  - policy
  - rbac.authorization.k8s.io
  - snapshot.storage.k8s.io
  - trident.netapp.io
  resources:
  - configmaps
  - cronjobs
  - daemonsets
  - deployments
```

```
  - horizontalpodautoscalers
  - ingresses
  - jobs
  - namespaces
  - networkpolicies
  - persistentvolumeclaims
  - poddisruptionbudgets
  - pods
  - podtemplates
  - replicasets
  - replicationcontrollers
  - replicationcontrollers/scale
  - rolebindings
  - roles
  - secrets
  - serviceaccounts
  - services
  - statefulsets
  - tridentmirrorrelationships
  - tridentsnapshotinfos
  - volumesnapshots
  - volumesnapshotcontents
  verbs:
  - delete

# Watch resources
# Necessary to monitor progress
- apiGroups:
  - ""
  resources:
  - pods
  - replicationcontrollers
  - replicationcontrollers/scale
  verbs:
  - watch

# Update resources
- apiGroups:
  - ""
  - build.openshift.io
  - image.openshift.io
  resources:
  - builds/details
  - replicationcontrollers
  - replicationcontrollers/scale
  - imagestreams/layers
```

```
  - imagestreamtags
  - imagetags
  verbs:
  - update
```

b. (Solo para clústeres de OpenShift) Añada lo siguiente al final del `astra-admin-account.yaml` archivo:

```
# OpenShift security
- apiGroups:
  - security.openshift.io
  resources:
  - securitycontextconstraints
  verbs:
  - use
  - update
```

c. Aplique el rol de clúster:

```
kubectl apply -f astra-admin-account.yaml
```

Rol del clúster ampliado

Este rol contiene permisos ampliados para que un clúster lo gestione Astra Control. Puedes usar este rol si utilizas varios contextos y no puedes utilizar el comando kubeconfig predeterminado de Astra Control configurado durante la instalación o un rol limitado con un único contexto no funcionará en tu entorno:

> Lo siguiente `ClusterRole` Los pasos son un ejemplo general de Kubernetes. Consulte la documentación de la distribución de Kubernetes para obtener instrucciones específicas de su entorno.

a. Cree un `ClusterRole` archivo llamado, por ejemplo, `astra-admin-account.yaml`.

```
<strong>astra-admin-account.yaml</strong>
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'
```

b. Aplique el rol de clúster:

```
kubectl apply -f astra-admin-account.yaml
```

3. Cree el enlace de rol de clúster para el rol del clúster a la cuenta de servicio:

a. Cree un `ClusterRoleBinding` archivo llamado `astracontrol-clusterrolebinding.yaml`.

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astracontrol-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: astra-admin-account
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

b. Aplique el enlace de roles del clúster:

```
kubectl apply -f astracontrol-clusterrolebinding.yaml
```

4. Crear y aplicar el secreto de token:
   a. Cree un archivo secreto de token llamado `secret-astracontrol-service-account.yaml`.

```
<strong>secret-astracontrol-service-account.yaml</strong>
```

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-astracontrol-service-account
  namespace: default
  annotations:
    kubernetes.io/service-account.name: "astracontrol-service-
account"
type: kubernetes.io/service-account-token
```

   b. Aplique el secreto de token:

```
kubectl apply -f secret-astracontrol-service-account.yaml
```

5. Agregue el secreto de token a la cuenta de servicio agregando su nombre a la `secrets` array (la última línea del siguiente ejemplo):

```
kubectl edit sa astracontrol-service-account
```

```
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-48xhx
kind: ServiceAccount
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"v1","kind":"ServiceAccount","metadata":{"annotations":{},
"name":"astracontrol-service-account","namespace":"default"}}
  creationTimestamp: "2023-06-14T15:25:45Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "2767069"
  uid: 2ce068c4-810e-4a96-ada3-49cbf9ec3f89
secrets:
- name: astracontrol-service-account-dockercfg-48xhx
<strong>- name: secret-astracontrol-service-account</strong>
```

6. Enumere los secretos de la cuenta de servicio, reemplazando `<context>` con el contexto correcto para su instalación:

```
kubectl get serviceaccount astracontrol-service-account --context
<context> --namespace default -o json
```

El final de la salida debe ser similar a lo siguiente:

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-48xhx"},
{ "name": "secret-astracontrol-service-account"}
]
```

Los índices de cada elemento de la `secrets` la matriz comienza con 0. En el ejemplo anterior, el índice para `astracontrol-service-account-dockercfg-48xhx` sería 0 y el índice para `secret-astracontrol-service-account` sería 1. En la salida, anote el número de índice del secreto de la cuenta de servicio. Necesitará este número de índice en el siguiente paso.

7. Genere la kubeconfig de la siguiente manera:
   a. Cree un `create-kubeconfig.sh` archivo.
   b. Sustituya `TOKEN_INDEX` al principio de la secuencia de comandos siguiente con el valor correcto.

```
<strong>create-kubeconfig.sh</strong>
```

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
```

```
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token-
user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

c. Origen de los comandos para aplicarlos al clúster de Kubernetes.

```
source create-kubeconfig.sh
```

8. (Opcional) cambie el nombre de la Marca de prestigio por un nombre significativo para el clúster.

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

(Vista previa técnica) Instale Astra Connector para clústeres gestionados

Los clústeres gestionados por Astra Control Center utilizan Astra Connector para permitir la comunicación entre el clúster gestionado y Astra Control Center. Debe instalar Astra Connector en todos los clústeres que desee gestionar.

Instala Astra Connector

Instalas Astra Connector con comandos de Kubernetes y archivos de recursos personalizados (CR).

Acerca de esta tarea

- Cuando realice estos pasos, ejecute estos comandos en el clúster que desee gestionar con Astra Control.
- Si utiliza un host de Bastion, emita estos comandos desde la línea de comandos del host de Bastion.

Antes de empezar

- Necesitas acceder al clúster que quieras gestionar con Astra Control.
- Necesitas permisos de administrador de Kubernetes para instalar el operador Astra Connector en el clúster.

Si el clúster está configurado con la aplicación de admisión de seguridad de POD, que es el valor predeterminado para los clústeres de Kubernetes 1,25 y posteriores, tiene que habilitar las restricciones PSA en los espacios de nombres correspondientes. Consulte "Prepare su entorno para la gestión de clústeres con Astra Control" si desea obtener instrucciones.

Pasos

1. Instala el operador Astra Connector en el clúster que quieras gestionar con Astra Control. Cuando se ejecuta este comando, el espacio de nombres `astra-connector-operator` se crea y la configuración se aplica al espacio de nombres:

```
kubectl apply -f https://github.com/NetApp/astra-connector-
operator/releases/download/24.02.0-
202403151353/astraconnector_operator.yaml
```

2. Compruebe que el operador está instalado y listo:

```
kubectl get all -n astra-connector-operator
```

3. Obtén un token de API de Astra Control. Consulte la "Documentación de Astra Automation" si desea obtener instrucciones.
4. Cree un secreto con el token. Reemplaza <API_TOKEN> por el token que has recibido de Astra Control:

```
kubectl create secret generic astra-token \
--from-literal=apiToken=<API_TOKEN> \
-n astra-connector
```

5. Crea un secreto de Docker para extraer la imagen de Astra Connector. Sustituya los valores entre paréntesis <> por información de su entorno:

Puedes encontrar la instancia de <ASTRA_CONTROL_ACCOUNT_ID> en la interfaz de usuario web de Astra Control. En la interfaz de usuario web, seleccione el icono de figura en la parte superior derecha de la página y seleccione **Acceso API**.

```
kubectl create secret docker-registry regcred \
--docker-username=<ASTRA_CONTROL_ACCOUNT_ID> \
--docker-password=<API_TOKEN> \
-n astra-connector \
--docker-server=cr.astra.netapp.io
```

6. Cree el archivo Astra Connector CR y asígnele el nombre `astra-connector-cr.yaml`. Actualiza los valores entre paréntesis <> para que coincidan con tu entorno de Astra Control y la configuración del clúster:
   - <ASTRA_CONTROL_ACCOUNT_ID>: Obtenida de la interfaz de usuario web de Astra Control durante el paso anterior.

- <CLUSTER_NAME>: El nombre que se debe asignar este clúster en Astra Control.
- <ASTRA_CONTROL_URL>: La URL de interfaz de usuario web de Astra Control. Por ejemplo:

```
https://astra.control.url
```

```
apiVersion: astra.netapp.io/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  astra:
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    clusterName: <CLUSTER_NAME>
    #Only set `skipTLSValidation` to `true` when using the default
self-signed
    #certificate in a proof-of-concept environment.
    skipTLSValidation: false #Should be set to false in production
environments
    tokenRef: astra-token
  natsSyncClient:
    cloudBridgeURL: <ASTRA_CONTROL_HOST_URL>
  imageRegistry:
    name: cr.astra.netapp.io
    secret: regcred
```

7. Después de rellenar el `astra-connector-cr.yaml` Con los valores correctos, aplique el CR:

```
kubectl apply -n astra-connector -f astra-connector-cr.yaml
```

8. Compruebe que Astra Connector está completamente implementado:

```
kubectl get all -n astra-connector
```

9. Compruebe que el clúster esté registrado en Astra Control:

```
kubectl get astraconnectors.astra.netapp.io -A
```

Debería ver una salida similar a la siguiente:

```
NAMESPACE          NAME               REGISTERED   ASTRACONNECTORID
STATUS
astra-connector    astra-connector    true         00ac8-2cef-41ac-8777-
ed0583e   Registered with Astra
```

10. Compruebe que el clúster aparezca en la lista de clústeres gestionados de la página **Clusters** de la interfaz de usuario web de Astra Control.

Añadir un clúster

Para comenzar a gestionar las aplicaciones, añada un clúster de Kubernetes y gestiónelo como un recurso de computación. Es necesario añadir un clúster para Astra Control Center para descubrir sus aplicaciones Kubernetes.

Le recomendamos que Astra Control Center gestione el clúster en el que se implementa primero antes de añadir otros clústeres a Astra Control Center para su gestión. Disponer del cluster inicial en administración es necesario para enviar datos Kubemetrics y datos asociados al cluster para mediciones y resolución de problemas.

Antes de empezar

- Antes de añadir un clúster, revise y realice la operación necesaria "requisitos previos".
- Si utiliza un controlador de SAN de ONTAP, asegúrese de que multivía esté habilitado en todos los clústeres de Kubernetes.

Pasos

1. Acceda desde el menú Dashboard o Clusters:
   - En **Panel** en Resumen de recursos, seleccione **Agregar** en el panel Clusters.
   - En el área de navegación de la izquierda, seleccione **Clusters** y, a continuación, seleccione **Add Cluster** en la página Clusters.
2. En la ventana **Agregar clúster** que se abre, cargue un `kubeconfig.yaml` archivar o pegar el contenido de un `kubeconfig.yaml` archivo.

   La `kubeconfig.yaml` el archivo debe incluir **sólo la credencial de cluster para un cluster**.

   Si crea el suyo propio `kubeconfig` file, debe definir sólo un elemento de contexto **uno** en él. Consulte "Documentación de Kubernetes" para obtener información acerca de cómo crear `kubeconfig` archivos. Si creó una imagen de kubeconfig para una función de clúster limitada mediante "este proceso", asegúrese de cargar o pegar esa kubeconfig en este paso.

3. Introduzca un nombre de credencial. De forma predeterminada, el nombre de las credenciales se completa automáticamente como nombre del clúster.
4. Seleccione **Siguiente**.
5. Seleccione la clase de almacenamiento predeterminada que se utilizará para este clúster de Kubernetes y seleccione **Siguiente**.

Debe seleccionar una clase de almacenamiento que esté configurada en el aprovisionador de control de Astra y que esté respaldada por el almacenamiento de ONTAP.

6. Revise la información y si todo parece bien, seleccione **Agregar**.

Resultado

El clúster entra en el estado **descubriendo** y luego cambia a **saludable**. Ahora está gestionando el clúster con Astra Control Center.

Después de agregar un clúster para administrarlo en Astra Control Center, puede que el operador de supervisión tarde unos minutos en implementar. Hasta entonces, el icono de notificación se vuelve rojo y registra un evento **Comprobación de estado del agente de supervisión fallida**. Puede ignorar esto porque el problema se resuelve cuando Astra Control Center obtiene el estado correcto. Si el problema no se resuelve en unos minutos, vaya al clúster y ejecute `oc get pods -n netapp-monitoring` como punto de partida. Tendrá que buscar en los registros del operador de supervisión para depurar el problema.

Habilite la autenticación en el back-end de almacenamiento ONTAP

El Centro de control de Astra ofrece dos modos de autenticación de un back-end de ONTAP:

- **Autenticación basada en credenciales**: El nombre de usuario y la contraseña de un usuario de ONTAP con los permisos requeridos. Debe utilizar un rol de inicio de sesión de seguridad predefinido, como admin o vsadmin, para garantizar la máxima compatibilidad con las versiones de ONTAP.
- **Autenticación basada en certificados**: Astra Control Center también puede comunicarse con un clúster de ONTAP utilizando un certificado instalado en el backend. Debe usar el certificado de cliente, la clave y el certificado de CA de confianza si se utilizan (recomendado).

Más adelante, puede actualizar los back-ends existentes para pasar de un tipo de autenticación a otro método. Solo se admite un método de autenticación a la vez.

Habilite la autenticación basada en credenciales

Astra Control Center requiere las credenciales para un ámbito del clúster `admin` Para comunicarse con el backend de ONTAP. Debe utilizar roles estándar predefinidos como `admin`. Esto garantiza la compatibilidad con futuras versiones de ONTAP que podrían exponer API de funciones para que las utilicen en futuras versiones del Centro de control de Astra.

Puede crearse y utilizarse un rol de inicio de sesión de seguridad personalizado con Astra Control Center, pero no es recomendable.

Una definición de backend de ejemplo tiene el siguiente aspecto:

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "admin",
  "password": "secret"
}
```

La definición de backend es el único lugar en el que las credenciales se almacenan en texto sin formato. La creación o actualización de un backend es el único paso que requiere conocimiento de las credenciales. De este modo, se trata de una operación exclusiva para administrador que realiza el administrador de Kubernetes o de almacenamiento.

Habilite la autenticación basada en certificados

Astra Control Center puede utilizar certificados para comunicarse con back-ends de ONTAP nuevos y existentes. Debe introducir la siguiente información en la definición de backend.

- `clientCertificate`: Certificado de cliente.
- `clientPrivateKey`: Clave privada asociada.
- `trustedCACertificate`: Certificado de CA de confianza. Si se utiliza una CA de confianza, se debe proporcionar este parámetro. Esto se puede ignorar si no se utiliza ninguna CA de confianza.

Es posible usar uno de los siguientes tipos de certificados:

- Certificado autofirmado
- Certificado de terceros

Habilite la autenticación con un certificado autofirmado

Un flujo de trabajo típico implica los pasos siguientes.

Pasos

1. Genere una clave y un certificado de cliente. Al generar, defina el nombre común (CN) en el usuario ONTAP para autenticarse como.

   ```
   openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
   -out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=<common-name>"
   ```

2. Instale el certificado de cliente de tipo `client-ca` Y el clúster de ONTAP.

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

3. Confirme que el rol de inicio de sesión de seguridad de ONTAP admite el método de autenticación de certificado.

```
security login create -user-or-group-name vsadmin -application ontapi
-authentication-method cert -vserver <vserver-name>
security login create -user-or-group-name vsadmin -application http
-authentication-method cert -vserver <vserver-name>
```

4. Pruebe la autenticación mediante el certificado generado. Sustituya <LIF de gestión de ONTAP> y <vserver name> por la IP de LIF de gestión y el nombre SVM. Debe asegurarse de que la LIF tiene su política de servicio establecida en `default-data-management`.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns=http://www.netapp.com/filer/admin version="1.21" vfiler="<vserver-
name>"><vserver-get></vserver-get></netapp>
```

5. Con los valores obtenidos del paso anterior, añada el back-end del almacenamiento en la interfaz de usuario de Astra Control Center.

Active la autenticación con un certificado de terceros

Si tiene un certificado de terceros, puede configurar la autenticación basada en certificados con estos pasos.

Pasos

1. Genere la clave privada y CSR:

```
openssl req -new -newkey rsa:4096 -nodes -sha256 -subj "/" -outform pem
-out ontap_cert_request.csr -keyout ontap_cert_request.key -addext
"subjectAltName = DNS:<ONTAP_CLUSTER_FQDN_NAME>,IP:<ONTAP_MGMT_IP>”
```

2. Transfiera la CSR a la CA de Windows (CA de terceros) y emita el certificado firmado.
3. Descargue el certificado firmado y asígnele el nombre `ontap_signed_cert.crt'
4. Exporte el certificado raíz de Windows CA (CA de terceros).
5. Asigne un nombre a este archivo `ca_root.crt`

   Ahora tiene los siguientes tres archivos:

- **Clave privada**: `ontap_signed_request.key` (Esta es la clave correspondiente para el certificado de servidor en ONTAP. Se necesita al instalar el certificado de servidor.)
- **Certificado firmado**: `ontap_signed_cert.crt` (Esto también se denomina *server certificate* en ONTAP.)
- **Certificado de CA raíz**: `ca_root.crt` (Esto también se denomina *server-ca certificate* en ONTAP.)

6. Instale estos certificados en ONTAP. Generar e instalar `server` y.. `server-ca` Certificados en ONTAP.

Expanda para sample.yaml

```
# Copy the contents of ca_root.crt and use it here.

security certificate install -type server-ca

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----


You should keep a copy of the CA-signed digital certificate for
future reference.

The installed certificate's CA and serial number for reference:

CA:
serial:

The certificate's generated name for reference:


===

# Copy the contents of ontap_signed_cert.crt and use it here. For
key, use the contents of ontap_cert_request.key file.
security certificate install -type server
Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----

Please enter Private Key: Press <Enter> when done

-----BEGIN PRIVATE KEY-----
<private key details>
-----END PRIVATE KEY-----

Enter certificates of certification authorities (CA) which form the
certificate chain of the server certificate. This starts with the
issuing CA certificate of the server certificate and can range up to
the root CA certificate.
Do you want to continue entering root and/or intermediate
```

```
certificates {y|n}: n

The provided certificate does not have a common name in the subject
field.
Enter a valid common name to continue installation of the
certificate: <ONTAP_CLUSTER_FQDN_NAME>

You should keep a copy of the private key and the CA-signed digital
certificate for future reference.
The installed certificate's CA and serial number for reference:
CA:
serial:
The certificate's generated name for reference:


==
# Modify the vserver settings to enable SSL for the installed
certificate

ssl modify -vserver <vserver_name> -ca <CA>  -server-enabled true
-serial <serial number>       (security ssl modify)

==
# Verify if the certificate works fine:

openssl s_client -CAfile ca_root.crt -showcerts -servername server
-connect <ONTAP_CLUSTER_FQDN_NAME>:443
CONNECTED(00000005)
depth=1 DC = local, DC = umca, CN = <CA>
verify return:1
depth=0
verify return:1
write W BLOCK
---
Certificate chain
0 s:
   i:/DC=local/DC=umca/<CA>

-----BEGIN CERTIFICATE-----
<Certificate details>
```

7. Cree el certificado de cliente para el mismo host para la comunicación sin contraseña. Astra Control Center utiliza este proceso para comunicarse con ONTAP.
8. Genere e instale los certificados de cliente en ONTAP:

Expanda para sample.yaml

```
# Use /CN=admin or use some other account which has privileges.
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout
ontap_test_client.key -out ontap_test_client.pem -subj "/CN=admin"

Copy the content of ontap_test_client.pem file and use it in the
below command:
security certificate install -type client-ca -vserver <vserver_name>

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<Certificate details>
-----END CERTIFICATE-----

You should keep a copy of the CA-signed digital certificate for
future reference.
The installed certificate's CA and serial number for reference:

CA:
serial:
The certificate's generated name for reference:


==

ssl modify -vserver <vserver_name> -client-enabled true
(security ssl modify)

# Setting permissions for certificates
security login create -user-or-group-name admin -application ontapi
-authentication-method cert -role admin -vserver <vserver_name>

security login create -user-or-group-name admin -application http
-authentication-method cert -role admin -vserver <vserver_name>

==

#Verify passwordless communication works fine with the use of only
certificates:

curl --cacert ontap_signed_cert.crt  --key ontap_test_client.key
--cert ontap_test_client.pem
https://<ONTAP_CLUSTER_FQDN_NAME>/api/storage/aggregates
{
```

```
"records": [
{
"uuid": "f84e0a9b-e72f-4431-88c4-4bf5378b41bd",
"name": "<aggr_name>",
"node": {
"uuid": "7835876c-3484-11ed-97bb-d039ea50375c",
"name": "<node_name>",
"_links": {
"self": {
"href": "/api/cluster/nodes/7835876c-3484-11ed-97bb-d039ea50375c"
}
}
},
"_links": {
"self": {
"href": "/api/storage/aggregates/f84e0a9b-e72f-4431-88c4-
4bf5378b41bd"
}
}
}
],
"num_records": 1,
"_links": {
"self": {
"href": "/api/storage/aggregates"
}
}
}%
```


9. Añada el back-end de almacenamiento en la interfaz de usuario de Astra Control Center y proporcione los siguientes valores:
   - **Certificado de cliente**: ontap_test_client.pem
   - **Clave privada**: ontap_test_client.key
   - **Certificado de CA de confianza**: ontap_signed_cert.crt

Añada un back-end de almacenamiento

Después de configurar las credenciales o la información de autenticación de certificados, puede añadir un back-end de almacenamiento de ONTAP existente a Astra Control Center para gestionar sus recursos.

Gestionar los clústeres de almacenamiento en Astra Control como back-end de almacenamiento le permite obtener vínculos entre los volúmenes persistentes (VP) y el back-end de almacenamiento, así como mediciones de almacenamiento adicionales.

Añadir y gestionar back-ends de almacenamiento de ONTAP en Astra Control Center es opcional cuando se

utiliza la tecnología SnapMirror de NetApp si has habilitado el aprovisionador de control de Astra.

Pasos

1. En el panel de control del área de navegación de la izquierda, seleccione **Backends**.
2. Seleccione **Agregar**.
3. En la sección Usar existente de la página Agregar backend de almacenamiento, seleccione **ONTAP**.
4. Seleccione una de las siguientes opciones:
   - **Usar credenciales de administrador**: Ingrese la dirección IP de administración del clúster de ONTAP y las credenciales de administración. Las credenciales deben ser credenciales para todo el clúster.

     El usuario cuyas credenciales introduzca aquí debe tener la `ontapi` Método de acceso de inicio de sesión de usuario habilitado en System Manager de ONTAP en el clúster de ONTAP. Si planea utilizar la replicación de SnapMirror, aplique las credenciales de usuario con el rol "admin", que tiene los métodos de acceso `ontapi` y.. `http`, En clústeres ONTAP de origen y destino. Consulte "Gestionar cuentas de usuario en la documentación de ONTAP" si quiere más información.

   - **Utilice un certificado**: Cargue el certificado `.pem` archivo, la clave de certificado `.key` archivo y, opcionalmente, el archivo de entidad de certificación.
5. Seleccione **Siguiente**.
6. Confirme los detalles del backend y seleccione **Administrar**.

Resultado

El back-end aparece en la `online` estado en la lista con información resumida.

Es posible que deba actualizar la página para que se muestre el back-end.

Añadir un bucket

Puede añadir un bloque con la interfaz de usuario de Astra Control o. "API de control Astra". Añadir proveedores de bloques de almacenamiento de objetos es esencial si desea realizar backups de sus aplicaciones y del almacenamiento persistente o si desea clonar aplicaciones entre clústeres. Astra Control almacena estas copias de seguridad o clones en los bloques de almacenamiento de objetos que defina.

No necesita un bloque de Astra Control si clona la configuración de sus aplicaciones y el almacenamiento persistente en el mismo clúster. La funcionalidad de snapshots de aplicaciones no requiere un bloque.

Antes de empezar

- Asegúrese de tener un bloque al que se puede acceder desde los clústeres que gestiona Astra Control Center.
- Asegúrese de tener credenciales para el bloque.
- Asegúrese de que el cucharón es uno de los siguientes tipos:
  - ONTAP S3 de NetApp
  - StorageGRID S3 de NetApp
  - Microsoft Azure

- Genérico S3

> Amazon Web Services (AWS) y Google Cloud Platform (GCP) utilizan el tipo de bloque Generic S3.

> Aunque Astra Control Center es compatible con Amazon S3 como proveedor de bloques Generic S3, es posible que Astra Control Center no admita todos los proveedores de almacenes de objetos que afirman que Amazon es compatible con S3.

Pasos

1. En el área de navegación de la izquierda, seleccione **Cuchos**.
2. Seleccione **Agregar**.
3. Seleccione el tipo de bloque.

> Cuando agregue un bloque, seleccione el proveedor de segmento correcto y proporcione las credenciales correctas para ese proveedor. Por ejemplo, la interfaz de usuario acepta ONTAP S3 de NetApp como tipo y acepta credenciales de StorageGRID; sin embargo, esto hará que se produzcan errores en todos los futuros backups de aplicaciones y restauraciones usando este bucket.

4. Introduzca un nombre de bloque existente y una descripción opcional.

> El nombre y la descripción del bloque aparecen como una ubicación de backup que se puede elegir más adelante al crear un backup. El nombre también aparece durante la configuración de la política de protección.

5. Introduzca el nombre o la dirección IP del extremo de S3.
6. En **Seleccionar credenciales**, elija la ficha **Agregar** o **utilizar existente**.
   - Si ha elegido **Agregar**:
     i. Introduzca un nombre para la credencial que la distingue de otras credenciales en Astra Control.
     ii. Escriba el identificador de acceso y la clave secreta pegando el contenido del portapapeles.
   - Si ha elegido **utilizar existente**:
     i. Seleccione las credenciales existentes que desea utilizar con el bloque.
7. Seleccione `Add`.

> Cuando se agrega un bloque, Astra Control Marca un bloque con el indicador de segmento predeterminado. El primer bloque que crea se convierte en el bloque predeterminado. A medida que se añaden bloques, más adelante se puede decidir a. "establecer otro bloque predeterminado".

Información de copyright

Información de la marca comercial