Agregue un clúster autogestionado

Astra Control Service

NetApp
April 24, 2024

Tabla de contenidos

Agregue un clúster autogestionado . . . 1
Añade un clúster público autogestionado a Astra Control Service . . . 1
Añade un clúster privado autogestionado a Astra Control Service . . . 5
Comprueba la versión de Astra Trident . . . 10
Cree un archivo kubeconfig . . . 12

Agregue un clúster autogestionado

Añade un clúster público autogestionado a Astra Control Service

Una vez configurado su entorno, estará listo para crear un clúster de Kubernetes y, a continuación, añadirlo al servicio Astra Control.

Un clúster autogestionado es un clúster que usted aprovisiona y gestiona directamente. Astra Control Service admite clústeres autogestionados que se ejecutan en un entorno de cloud público. Puede agregar un clúster autogestionado a Astra Control Service cargando un `kubeconfig.yaml` archivo. Deberá asegurarse de que el clúster cumple los requisitos que se indican aquí.

Distribuciones de Kubernetes compatibles

Puede utilizar Astra Control Service para gestionar los siguientes tipos de clústeres públicos autogestionados:

Distribución de Kubernetes	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

En estas instrucciones se asume que ya ha creado un clúster autogestionado.

- Añada el clúster a Astra Control Service
- Cambie la clase de almacenamiento predeterminada

Añada el clúster a Astra Control Service

Después de iniciar sesión en Astra Control Service, el primer paso es empezar a gestionar los clústeres. Antes de añadir un clúster al servicio Astra Control Service, tendrá que realizar tareas específicas y asegurarse de que el clúster cumple determinados requisitos.

Antes de empezar

Un clúster autogestionado es un clúster que usted aprovisiona y gestiona directamente. Astra Control Service admite clústeres autogestionados que se ejecutan en un entorno de cloud público. Los clústeres autogestionados pueden utilizar Astra Control Provisioner para interactuar con los servicios de almacenamiento de NetApp o pueden usar controladores de la interfaz de almacenamiento de contenedores (CSI) para interactuar con Amazon Elastic Block Store (EBS), los discos gestionados de Azure y el disco persistente de Google.

Astra Control Service es compatible con clústeres autogestionados que utilizan las siguientes distribuciones de Kubernetes:

- OpenShift Container Platform de Red Hat
- Motor Kubernetes del rancher
- Subida de Kubernetes

Su clúster autogestionado debe cumplir con los siguientes requisitos:

- El clúster debe estar accesible a través de Internet.
- Si está utilizando o planea utilizar almacenamiento habilitado con controladores CSI, se deben instalar los controladores CSI adecuados en el clúster. Para obtener más información sobre el uso de los controladores CSI para integrar el almacenamiento, consulte la documentación del servicio de almacenamiento.
- Tiene acceso al archivo kubeconfig de cluster que incluye solo un elemento de contexto. Siga "estas instrucciones" para generar un archivo kubeconfig.
- Si va a agregar el clúster mediante un archivo kubeconfig que hace referencia a una entidad de certificación (CA) privada, agregue la siguiente línea al `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.

Pasos

1. En el Panel de control, seleccione **gestionar clúster Kubernetes**.

   Siga las instrucciones para añadir el clúster.

2. **Proveedor**: Selecciona la pestaña **Otro** para agregar detalles sobre tu cluster autogestionado.
   a. **Otros**: Proporcione detalles sobre su cluster autogestionado cargando un `kubeconfig.yaml` o bien, pegue el contenido de `kubeconfig.yaml` desde el portapapeles.

   > 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.

3. **Nombre de credencial**: Proporciona un nombre para la credencial de clúster autogestionada que estás cargando en Astra Control. De forma predeterminada, el nombre de las credenciales se completa automáticamente como nombre del clúster.
4. **Identificador de ruta privado**: Este campo es solo para uso con clusters privados.
5. Seleccione **Siguiente**.
6. (Opcional) **Almacenamiento**: Opcionalmente, seleccione la clase de almacenamiento que desea que las aplicaciones de Kubernetes implementadas en este clúster utilicen de forma predeterminada.
   a. Para seleccionar una nueva clase de almacenamiento predeterminada para el clúster, active la casilla de verificación **Asignar una nueva clase de almacenamiento predeterminada**.
   b. Seleccione una nueva clase de almacenamiento predeterminada de la lista.

   > Cada servicio de almacenamiento de proveedor de cloud muestra la siguiente información sobre el precio, el rendimiento y la resiliencia:
   >
   > - Cloud Volumes Service para Google Cloud: Información de precio, rendimiento y resiliencia
   > - Google Persistent Disk: No hay información de precio, rendimiento ni resiliencia disponible
   > - Azure NetApp Files: Información sobre rendimiento y resiliencia
   > - Discos administrados de Azure: No hay información de precios, rendimiento ni resiliencia disponible
   > - Amazon Elastic Block Store: No dispone de información de precio, rendimiento o resiliencia
   > - Amazon FSX para ONTAP de NetApp: Sin información de precio, rendimiento ni resiliencia disponible
   > - Cloud Volumes ONTAP de NetApp: No hay información de precio, rendimiento ni resiliencia disponible

Cada clase de almacenamiento puede utilizar uno de los siguientes servicios:

- "Cloud Volumes Service para Google Cloud"
- "Disco persistente de Google"
  - "Azure NetApp Files"
  - "Discos gestionados de Azure"
  - "Amazon Elastic Block Store"
  - "Amazon FSX para ONTAP de NetApp"
  - "Cloud Volumes ONTAP de NetApp"

    Más información acerca de "Clases de almacenamiento para clústeres de Amazon Web Services". Más información acerca de "Clases de almacenamiento para clústeres de AKS". Más información acerca de "Clases de almacenamiento para clústeres GKE".

c. Seleccione **Siguiente**.

d. **Revisar y aprobar**: Revise los detalles de la configuración.

e. Selecciona **Add** para agregar el clúster a Astra Control Service.

Cambie la clase de almacenamiento predeterminada

Es posible cambiar la clase de almacenamiento predeterminada para un clúster de.

Cambie la clase de almacenamiento predeterminada con Astra Control

Puede cambiar la clase de almacenamiento predeterminada para un clúster de Astra Control. Si su clúster utiliza un servicio de fondo de almacenamiento previamente instalado, es posible que no pueda utilizar este método para cambiar la clase de almacenamiento predeterminada (la acción **establecer como predeterminada** no se puede seleccionar). En este caso, usted puede Cambie la clase de almacenamiento predeterminada con la línea de comandos.

Pasos

1. En la interfaz de usuario de Astra Control Service, seleccione **Clusters**.
2. En la página **Clusters**, seleccione el clúster que desea cambiar.
3. Seleccione la ficha **almacenamiento**.
4. Seleccione la categoría **clases de almacenamiento**.
5. Seleccione el menú **acciones** para la clase de almacenamiento que desea establecer como predeterminada.
6. Seleccione **establecer como predeterminado**.

Cambie la clase de almacenamiento predeterminada con la línea de comandos

Es posible cambiar la clase de almacenamiento predeterminada para un clúster mediante comandos de Kubernetes. Este método funciona independientemente de la configuración del clúster.

Pasos

1. Inicie sesión en su clúster de Kubernetes.
2. Enumere las clases de almacenamiento del clúster:

```
kubectl get storageclass
```

3. Quite la designación predeterminada de la clase de almacenamiento predeterminada. Sustituya <SC_NAME> por el nombre de la clase de almacenamiento:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Seleccione una clase de almacenamiento diferente de forma predeterminada. Sustituya <SC_NAME> por el nombre de la clase de almacenamiento:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confirme la nueva clase de almacenamiento predeterminada:

```
kubectl get storageclass
```

Añade un clúster privado autogestionado a Astra Control Service

Una vez configurado su entorno, estará listo para crear un clúster de Kubernetes y, a continuación, añadirlo al servicio Astra Control.

Un clúster autogestionado es un clúster que usted aprovisiona y gestiona directamente. Astra Control Service admite clústeres autogestionados que se ejecutan en un entorno de cloud público. Puede agregar un clúster autogestionado a Astra Control Service cargando un `kubeconfig.yaml` archivo. Deberá asegurarse de que el clúster cumple los requisitos que se indican aquí.

Distribuciones de Kubernetes compatibles

Puede utilizar Astra Control Service para gestionar los siguientes tipos de clústeres privados autogestionados:

Distribución de Kubernetes	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

Distribución de Kubernetes	Versiones compatibles
OpenShift Container Platform de Red Hat	4,12 hasta 4,14

En estas instrucciones se asume que ya ha creado un clúster privado y ha preparado un método seguro para acceder de forma remota a él.

Tienes que realizar las siguientes tareas para añadir tu clúster privado a Astra Control Service:

1. Instala Astra Connector
2. Configure el almacenamiento persistente
3. Añada el clúster autogestionado privado a Astra Control Service

Instala Astra Connector

Antes de agregar un clúster privado, tiene que instalar Astra Connector en el clúster para que Astra Control se pueda comunicar con él. Consulte "Instala la versión anterior de Astra Connector para clústeres privados gestionados con flujos de trabajo que no sean nativos de Kubernetes" si desea obtener instrucciones.

Configure el almacenamiento persistente

Configure el almacenamiento persistente para el clúster. Consulte la documentación para empezar para obtener más información sobre la configuración del almacenamiento persistente:

- "Configure Microsoft Azure con Azure NetApp Files"
- "Configure Microsoft Azure con discos gestionados de Azure"
- "Configure Amazon Web Services"
- "Configure Google Cloud"

Añada el clúster autogestionado privado a Astra Control Service

Ahora puede añadir el clúster privado a Astra Control Service.

Antes de empezar

Un clúster autogestionado es un clúster que usted aprovisiona y gestiona directamente. Astra Control Service admite clústeres autogestionados que se ejecutan en un entorno de cloud público. Los clústeres autogestionados pueden utilizar Astra Control Provisioner para interactuar con los servicios de almacenamiento de NetApp o pueden usar controladores de la interfaz de almacenamiento de contenedores (CSI) para interactuar con Amazon Elastic Block Store (EBS), los discos gestionados de Azure y el disco persistente de Google.

Astra Control Service es compatible con clústeres autogestionados que utilizan las siguientes distribuciones de Kubernetes:

- OpenShift Container Platform de Red Hat
- Motor Kubernetes del rancher
- Subida de Kubernetes

Su clúster autogestionado debe cumplir con los siguientes requisitos:

- El clúster debe estar accesible a través de Internet.
- Si está utilizando o planea utilizar almacenamiento habilitado con controladores CSI, se deben instalar los controladores CSI adecuados en el clúster. Para obtener más información sobre el uso de los controladores CSI para integrar el almacenamiento, consulte la documentación del servicio de almacenamiento.
- Tiene acceso al archivo kubeconfig de cluster que incluye solo un elemento de contexto. Siga "estas instrucciones" para generar un archivo kubeconfig.
- Si va a agregar el clúster mediante un archivo kubeconfig que hace referencia a una entidad de certificación (CA) privada, agregue la siguiente línea al `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.

Pasos

1. En el Panel de control, seleccione **gestionar clúster Kubernetes**.

   Siga las instrucciones para añadir el clúster.

2. **Proveedor**: Selecciona la pestaña **Otro** para agregar detalles sobre tu cluster autogestionado.
3. **Otros**: Proporcione detalles sobre su cluster autogestionado cargando un `kubeconfig.yaml` o bien, pegue el contenido de `kubeconfig.yaml` desde el portapapeles.

   > Si crea el suyo propio `kubeconfig` file, debe definir sólo un elemento de contexto **uno** en él. Consulte "estas instrucciones" para obtener información acerca de cómo crear `kubeconfig` archivos.

4. **Nombre de credencial**: Proporciona un nombre para la credencial de clúster autogestionada que estás cargando en Astra Control. De forma predeterminada, el nombre de las credenciales se completa automáticamente como nombre del clúster.
5. **Identificador de ruta privado**: Introduce el identificador de ruta privado, que puedes obtener del Astra Connector. Si consulta el Astra Connector a través del `kubectl get astraconnector -n astra-connector` comando, el identificador de ruta privada se denomina `ASTRACONNECTORID`.

   > El identificador de ruta privada es el nombre asociado con Astra Connector que permite gestionar un clúster de Kubernetes privado con Astra. En este contexto, un clúster privado es un clúster de Kubernetes que no expone su servidor API a Internet.

6. Seleccione **Siguiente**.
7. (Opcional) **Almacenamiento**: Opcionalmente, seleccione la clase de almacenamiento que desea que las aplicaciones de Kubernetes implementadas en este clúster utilicen de forma predeterminada.
   a. Para seleccionar una nueva clase de almacenamiento predeterminada para el clúster, active la casilla de verificación **Asignar una nueva clase de almacenamiento predeterminada**.
   b. Seleccione una nueva clase de almacenamiento predeterminada de la lista.

Cada servicio de almacenamiento de proveedor de cloud muestra la siguiente información sobre el precio, el rendimiento y la resiliencia:

- Cloud Volumes Service para Google Cloud: Información de precio, rendimiento y resiliencia
- Google Persistent Disk: No hay información de precio, rendimiento ni resiliencia disponible
- Azure NetApp Files: Información sobre rendimiento y resiliencia
- Discos administrados de Azure: No hay información de precios, rendimiento ni resiliencia disponible
- Amazon Elastic Block Store: No dispone de información de precio, rendimiento o resiliencia
- Amazon FSX para ONTAP de NetApp: Sin información de precio, rendimiento ni resiliencia disponible
- Cloud Volumes ONTAP de NetApp: No hay información de precio, rendimiento ni resiliencia disponible

Cada clase de almacenamiento puede utilizar uno de los siguientes servicios:

- "Cloud Volumes Service para Google Cloud"
- "Disco persistente de Google"
- "Azure NetApp Files"
- "Discos gestionados de Azure"
- "Amazon Elastic Block Store"
- "Amazon FSX para ONTAP de NetApp"
- "Cloud Volumes ONTAP de NetApp"

  Más información acerca de "Clases de almacenamiento para clústeres de Amazon Web Services". Más información acerca de "Clases de almacenamiento para clústeres de AKS". Más información acerca de "Clases de almacenamiento para clústeres GKE".

c. Seleccione **Siguiente**.

d. **Revisar y aprobar**: Revise los detalles de la configuración.

e. Selecciona **Add** para agregar el clúster a Astra Control Service.

Cambie la clase de almacenamiento predeterminada

Es posible cambiar la clase de almacenamiento predeterminada para un clúster de.

Cambie la clase de almacenamiento predeterminada con Astra Control

Puede cambiar la clase de almacenamiento predeterminada para un clúster de Astra Control. Si su clúster utiliza un servicio de fondo de almacenamiento previamente instalado, es posible que no pueda utilizar este método para cambiar la clase de almacenamiento predeterminada (la acción **establecer como predeterminada** no se puede seleccionar). En este caso, usted puede Cambie la clase de almacenamiento predeterminada con la línea de comandos.

Pasos

1. En la interfaz de usuario de Astra Control Service, seleccione **Clusters**.
2. En la página **Clusters**, seleccione el clúster que desea cambiar.
3. Seleccione la ficha **almacenamiento**.
4. Seleccione la categoría **clases de almacenamiento**.
5. Seleccione el menú **acciones** para la clase de almacenamiento que desea establecer como predeterminada.
6. Seleccione **establecer como predeterminado**.

Cambie la clase de almacenamiento predeterminada con la línea de comandos

Es posible cambiar la clase de almacenamiento predeterminada para un clúster mediante comandos de Kubernetes. Este método funciona independientemente de la configuración del clúster.

Pasos

1. Inicie sesión en su clúster de Kubernetes.
2. Enumere las clases de almacenamiento del clúster:

```
kubectl get storageclass
```

3. Quite la designación predeterminada de la clase de almacenamiento predeterminada. Sustituya <SC_NAME> por el nombre de la clase de almacenamiento:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Seleccione una clase de almacenamiento diferente de forma predeterminada. Sustituya <SC_NAME> por el nombre de la clase de almacenamiento:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confirme la nueva clase de almacenamiento predeterminada:

```
kubectl get storageclass
```

Comprueba la versión de Astra Trident

Para añadir un clúster autogestionado que utilice el aprovisionador de Astra Control o Astra Trident para los servicios de almacenamiento, asegúrese de que la versión instalada de Astra Trident sea la versión 23,10 o la más reciente.

Pasos

1. Determine la versión de Astra Trident que ejecuta:

```
kubectl get tridentversions -n trident
```

Si Astra Trident está instalado, verá una salida similar a la siguiente:

```
NAME      VERSION
trident   24.02.0
```

Si Astra Trident no está instalado, verá una salida similar a la 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 los pods estén ejecutando:

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

4. Compruebe si las clases de almacenamiento utilizan 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
```

Cree un archivo kubeconfig

Puede añadir un clúster a Astra Control Service mediante un archivo kubeconfig. En función del tipo de cluster que desee agregar, es posible que necesite crear manualmente un archivo kubeconfig para el cluster mediante pasos específicos.

- Cree un archivo kubeconfig para los clústeres de Amazon EKS
- Cree un archivo kubeconfig para los clústeres de Red Hat OpenShift Service en AWS (ROSA)
- Cree un archivo kubeconfig para otros tipos de clusters

Cree un archivo kubeconfig para los clústeres de Amazon EKS

Siga estas instrucciones para crear un archivo kubeconfig y un secreto de token permanente para los clústeres de Amazon EKS. Se necesita un secreto de token permanente para los clústeres alojados en EKS.

Pasos

1. Siga las instrucciones de la documentación de Amazon para generar un archivo kubeconfig:

   "Creación o actualización de un archivo kubeconfig para un clúster de Amazon EKS"

2. Cree una cuenta de servicio del siguiente modo:

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

      Ajuste el nombre de la cuenta de servicio según sea necesario. El espacio de nombres `kube-system` es necesario para estos pasos. Si cambia aquí el nombre de la cuenta de servicio, debe aplicar los mismos cambios en los siguientes pasos.

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

+

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astra-admin-account
  namespace: kube-system
```

3. Aplicar la cuenta de servicio:

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

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

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

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astra-admin-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: astra-admin-account
  namespace: kube-system
```

5. Aplique el enlace de roles del clúster:

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

6. Se ha llamado a crear un archivo secreto de token de cuenta de servicio `astracontrol-secret.yaml`.

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

```yaml
apiVersion: v1
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: astra-admin-account
  name: astra-admin-account
  namespace: kube-system
type: kubernetes.io/service-account-token
```

7. Aplique el secreto de token:

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

8. Recupere el secreto de token:

```
kubectl get secret astra-admin-account -n kube-system -o
jsonpath='{.data.token}' | base64 -d
```

9. Sustituya el `user` Sección del archivo kubeconfig de AWS EKS con el token, como se muestra en el siguiente ejemplo:

```
user:
    token: k8s-aws-
v1.aHR0cHM6Ly9zdHMudXMtd2VzdC0yLmFtYXpvbmF3cy5jb20vP0FjdGlvbj1HZXRDYWxsZ
XJJZGVudGl0eSZWZXJzaW9uPTIwMTEtMDYtMTUmWC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQ
y1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBM1JEWDdKU0haWU9LSEQ2SyUyRjIwMjMwN
DAzJTJGdXMtd2VzdC0yJTJGc3RzJTJGYXdzNF9yZXF1ZXN0JlgtQW16LURhdGU9MjAyMzA0M
DNUMjA0MzQwWiZYLUFtei1FeHBpcmVzPTYwJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCUzQ
ngtazhzLWF3cy1pZCZYLUFtei1TaWduYXR1cmU9YjU4ZWM0NzdiM2NkZGYxNGRhNzU4MGI2Z
WQ2zY2NzI2YWIwM2UyNThjMjRhNTJjNmVhNjc4MTRlNjJkOTg2Mg
```

Cree un archivo kubeconfig para los clústeres de Red Hat OpenShift Service en AWS (ROSA)

Siga estas instrucciones para crear un archivo kubeconfig para clústeres de Red Hat OpenShift Service en AWS (ROSA).

Pasos

1. Inicie sesión en el clúster ROSA.
2. Cree una cuenta de servicio:

```
oc create sa astracontrol-service-account
```

3. Añada un rol de clúster:

```
oc adm policy add-cluster-role-to-user cluster-admin -z astracontrol-
service-account
```

4. Con el siguiente ejemplo, cree un archivo de configuración secreto de cuenta de servicio:

```
<strong>secret-astra-sa.yaml</strong>
```

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

5. Cree el secreto:

```
oc create -f secret-astra-sa.yaml
```

6. Edite la cuenta de servicio que ha creado y agregue el nombre secreto de la cuenta de servicio de Astra Control a `secrets` sección:

```
oc edit sa astracontrol-service-account
```

```yaml
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-dvfcd
kind: ServiceAccount
metadata:
  creationTimestamp: "2023-08-04T04:18:30Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "169770"
  uid: 965fa151-923f-4fbd-9289-30cad15998ac
secrets:
- name: astracontrol-service-account-dockercfg-dvfcd
- name: secret-astracontrol-service-account ####ADD THIS ONLY####
```

7. 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-dvfcd"},
{ "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-dvfcd` 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.

8. Genere la kubeconfig de la siguiente manera:

   a. Cree un `create-kubeconfig.sh` archivo. 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
```

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

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

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

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

Cree un archivo kubeconfig para otros tipos de clusters

Siga estas instrucciones para crear un archivo kubeconfig de rol limitado o ampliado para Rancher, upstream Kubernetes y Red Hat OpenShift clusters.

Para los clústeres que se gestionan mediante kubeconfig, opcionalmente puede crear un rol de administrador de permisos limitado o de permisos ampliados para Astra Control Service.

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:

- A. "versión compatible" de kubectl está instalado.
- Acceso kubectl al clúster que pretendes añadir y gestionar mediante Astra Control Service

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

- 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>
```

```yaml
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>
```

```yaml
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>
```

```yaml
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>
```

```yaml
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>
```

```yaml
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
```