Proteja sus aplicaciones

Astra Control Center

NetApp
June 07, 2024

Tabla de contenidos

Proteja sus aplicaciones

Información general sobre la protección

Puede crear backups, clones, snapshots y políticas de protección para sus aplicaciones con Astra Control Center. El backup de sus aplicaciones ayuda a que los servicios y los datos asociados estén disponibles lo más posible; durante un desastre, la restauración a partir de una copia de seguridad puede garantizar la recuperación completa de una aplicación y sus datos asociados con una interrupción mínima. Los backups, clones y copias Snapshot pueden ayudar a protegerse frente a amenazas comunes como el ransomware, la pérdida accidental de datos y los desastres medioambientales. "Conozca los tipos disponibles de protección de datos en Astra Control Center y cuándo utilizarlas".

Flujo de trabajo de protección de aplicaciones

Puede utilizar el siguiente ejemplo de flujo de trabajo para empezar a proteger las aplicaciones.

[Uno] Realice copias de seguridad de todas las aplicaciones

Para asegurarse de que sus aplicaciones están protegidas inmediatamente, "cree una copia de seguridad manual de todas las aplicaciones".

[Dos] Configure una política de protección para cada aplicación

Para automatizar futuros backups y copias Snapshot, "configure una política de protección para cada aplicación". A modo de ejemplo, puede comenzar con backups semanales y snapshots diarias, con una retención de un mes para ambos. La automatización de backups y snapshots con una política de protección es muy recomendada con respecto a copias de Snapshot y backups manuales.

[Tres] Opcional: Ajuste las políticas de protección

A medida que cambian las aplicaciones y sus patrones de uso, ajuste las políticas de protección según sea necesario para proporcionar la mejor protección.

[Cuatro] En caso de desastre, restaure sus aplicaciones

Si se produce la pérdida de datos, puede recuperarlo "restaurar la copia de seguridad más reciente" la primera para cada aplicación. Luego puede restaurar la snapshot más reciente (si está disponible).

Proteja las aplicaciones con snapshots y backups

Proteja sus aplicaciones tomando snapshots y backups usando una política de protección automatizada o de manera ad hoc. Puede utilizar la interfaz de usuario de Astra o. "La API de control Astra" para proteger aplicaciones.

> Si utiliza Helm para implantar aplicaciones, Astra Control Center requiere Helm versión 3. Se admite por completo la gestión y clonación de las aplicaciones implementadas con Helm 3 (o actualizadas de Helm 2 a Helm 3). Las aplicaciones implementadas con Helm 2 no son compatibles.

Al crear un proyecto para alojar una aplicación en un clúster de OpenShift, se asigna un UID de SecurityContext al proyecto (o espacio de nombres de Kubernetes). Para habilitar Astra Control Center para proteger su aplicación y mover la aplicación a otro clúster o proyecto en OpenShift, debe agregar directivas que permitan que la aplicación se ejecute como cualquier UID. Por ejemplo, los siguientes comandos de la CLI de OpenShift otorgan las directivas adecuadas a una aplicación de WordPress.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

Configure una política de protección

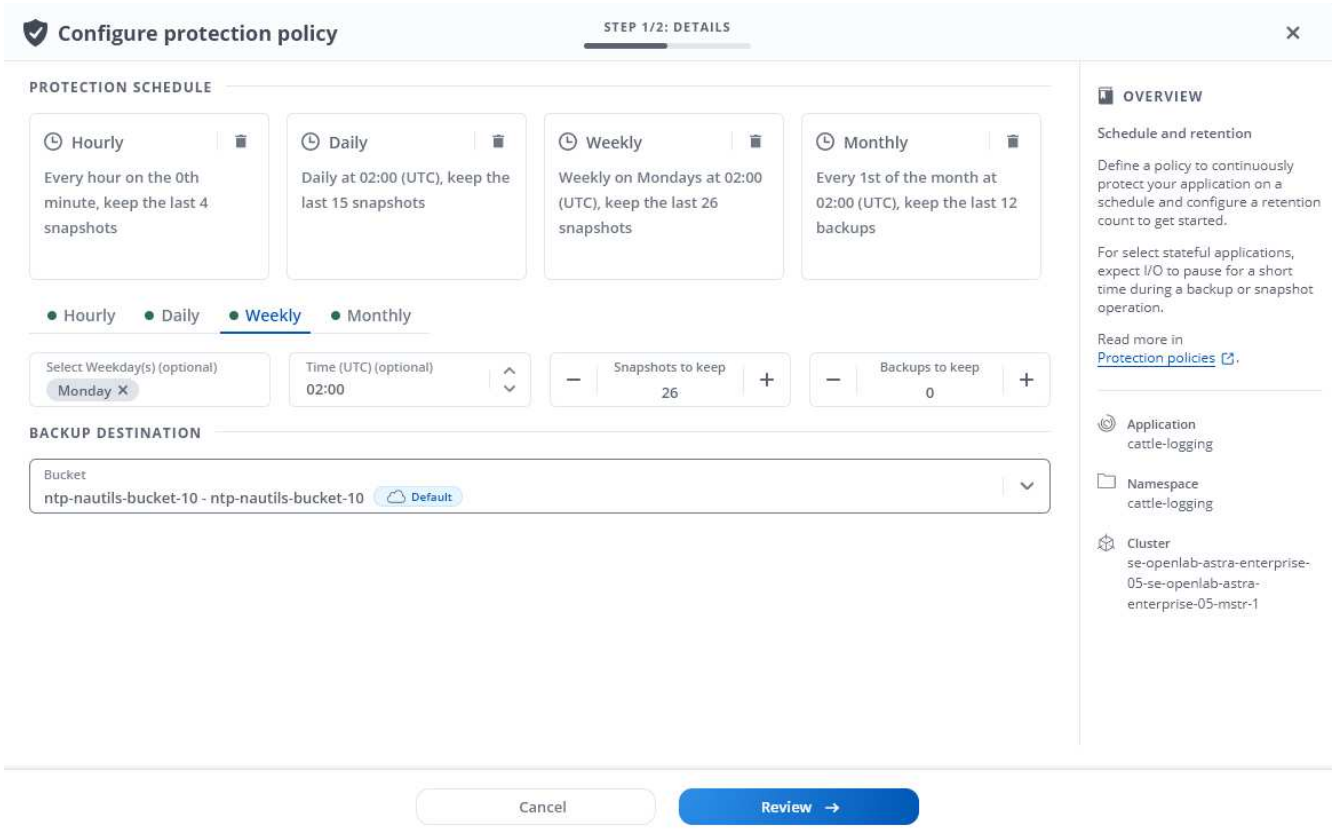
La política de protección protege una aplicación mediante la creación de snapshots, backups o ambos con una programación definida. Puede optar por crear snapshots y backups por hora, día, semana y mes, y especificar la cantidad de copias que desea retener. A modo de ejemplo, una política de protección puede crear backups semanales y copias Snapshot diarias, y conservar los backups y las copias Snapshot por un mes. La frecuencia con la que se crean snapshots y backups y el tiempo que se retienen depende de las necesidades de la organización.

Pasos

1. Seleccione **aplicaciones** y, a continuación, seleccione el nombre de una aplicación.
2. Seleccione **Protección de datos**.
3. Seleccione **Configurar política de protección**.
4. Defina una programación de protección eligiendo la cantidad de snapshots y backups que se mantendrán por hora, día, semana y mes.

   Puede definir las programaciones por hora, por día, por semana y por mes de forma simultánea. Una programación no se activa hasta que se establece un nivel de retención.

   En el siguiente ejemplo, se establecen cuatro programaciones de protección: Por hora, día, semana y mes para las copias Snapshot y los backups.



5. Seleccione **Revisión**.
6. Seleccione **Configurar política de protección.**

Resultado

Astra Control Center implementa la normativa de protección de datos mediante la creación y retención de instantáneas y copias de seguridad con la programación y retención que ha definido.

Crear una copia de Snapshot

Puede crear una snapshot bajo demanda en cualquier momento.

Pasos

1. Seleccione **aplicaciones**.
2. En el menú Opciones de la columna **acciones** de la aplicación deseada, seleccione **Snapshot**.
3. Personalice el nombre de la instantánea y, a continuación, seleccione **Revisión**.
4. Revise el resumen de la instantánea y seleccione **Snapshot**.

Resultado

Se inicia el proceso Snapshot. Una instantánea se realiza correctamente cuando el estado es **disponible** en la columna **acciones** de la página **Protección de datos** > **instantáneas**.

Cree un backup

También puede realizar copias de seguridad de una aplicación en cualquier momento.

Los bloques de S3 de Astra Control Center no informan sobre la capacidad disponible. Antes de realizar una copia de seguridad o clonar aplicaciones gestionadas por Astra Control Center, compruebe la información de los bloques en el sistema de gestión ONTAP o StorageGRID.

Pasos

1. Seleccione **aplicaciones**.
2. En el menú Opciones de la columna **acciones** de la aplicación deseada, seleccione **copia de seguridad**.
3. Personalice el nombre del backup.
4. Elija si desea realizar una copia de seguridad de la aplicación desde una instantánea existente. Si selecciona esta opción, puede elegir entre una lista de snapshots existentes.
5. Seleccione un destino para el backup seleccionando de la lista de bloques de almacenamiento.
6. Seleccione **Revisión**.
7. Revise el resumen de copia de seguridad y seleccione **copia de seguridad**.

Resultado

Astra Control Center crea una copia de seguridad de la aplicación.

Si la red tiene una interrupción del servicio o es anormalmente lenta, es posible que se agote el tiempo de espera de una operación de backup. Esto provoca un error en el backup.

No existe ninguna forma de detener un backup en ejecución. Si necesita eliminar el backup, espere hasta que se haya completado y, a continuación, utilice las instrucciones de Eliminar backups. Para eliminar una copia de seguridad fallida, "Utilice la API Astra Control".

Después de una operación de protección de datos (clonado, backup, restauración) y un cambio de tamaño posterior de volumen persistente, se demora hasta veinte minutos antes de que se muestre el tamaño del nuevo volumen en la interfaz de usuario. La operación de protección de datos se realiza correctamente en cuestión de minutos, y se puede utilizar el software de gestión para el back-end de almacenamiento para confirmar el cambio de tamaño del volumen.

Ver Snapshot y backups

Puede ver las instantáneas y las copias de seguridad de una aplicación desde la pestaña Data Protection.

Pasos

1. Seleccione **aplicaciones** y, a continuación, seleccione el nombre de una aplicación.
2. Seleccione **Protección de datos**.

   Las instantáneas se muestran de forma predeterminada.

3. Seleccione **copias de seguridad** para ver la lista de copias de seguridad.

Eliminar snapshots

Elimine las snapshots programadas o bajo demanda que ya no necesite.

Pasos

1. Seleccione **aplicaciones** y, a continuación, seleccione el nombre de una aplicación.
2. Seleccione **Protección de datos**.
3. En el menú Opciones de la columna **acciones** de la instantánea deseada, seleccione **Eliminar instantánea**.
4. Escriba la palabra "delete" para confirmar la eliminación y, a continuación, seleccione **Yes, Delete snapshot**.

Resultado

Astra Control Center elimina la instantánea.

Eliminar backups

Elimine los backups programados o bajo demanda que ya no necesita.

> No existe ninguna forma de detener un backup en ejecución. Si necesita eliminar el backup, espere hasta que se haya completado y, a continuación, utilice estas instrucciones. Para eliminar una copia de seguridad fallida, "Utilice la API Astra Control".

1. Seleccione **aplicaciones** y, a continuación, seleccione el nombre de una aplicación.
2. Seleccione **Protección de datos**.
3. Seleccione **copias de seguridad**.
4. En el menú Opciones de la columna **acciones** de la copia de seguridad deseada, seleccione **Eliminar copia de seguridad**.
5. Escriba la palabra "delete" para confirmar la eliminación y, a continuación, seleccione **Yes, Delete backup**.

Resultado

Astra Control Center elimina la copia de seguridad.

Restaurar aplicaciones

Astra Control puede restaurar su aplicación a partir de una instantánea o una copia de seguridad. La restauración a partir de una snapshot existente será más rápida cuando se restaure la aplicación en el mismo clúster. Puede utilizar la interfaz de usuario de Astra Control o. "La API de control Astra" para restaurar aplicaciones.

Acerca de esta tarea

- Se recomienda tomar una instantánea o realizar una copia de seguridad de la aplicación antes de restaurarla. Esto le permitirá clonar desde la snapshot o backup en el caso de que la restauración no se realice correctamente.
- Si utiliza Helm para implantar aplicaciones, Astra Control Center requiere Helm versión 3. Se admite por completo la gestión y clonación de las aplicaciones implementadas con Helm 3 (o actualizadas de Helm 2 a Helm 3). Las aplicaciones implementadas con Helm 2 no son compatibles.
- Si restaura en un clúster diferente, asegúrese de que el clúster utilice el mismo modo de acceso de volumen persistente (por ejemplo, ReadWriteMany). Se producirá un error en la operación de restauración si el modo de acceso al volumen persistente de destino es diferente.
- Cualquier usuario miembro con restricciones de espacio de nombres por nombre/ID de espacio de nombres o por etiquetas de espacio de nombres puede clonar o restaurar una aplicación en un nuevo

espacio de nombres en el mismo clúster o en cualquier otro clúster de la cuenta de su organización. Sin embargo, el mismo usuario no puede acceder a la aplicación clonada o restaurada en el nuevo espacio de nombres. Después de crear un espacio de nombres nuevo mediante una operación de clonado o restauración, el propietario/administrador de la cuenta puede editar las restricciones de la cuenta de usuario miembro y actualizar las restricciones de roles para que el usuario afectado conceda acceso al nuevo espacio de nombres.

- Al crear un proyecto para alojar una aplicación en un clúster de OpenShift, se asigna un UID de SecurityContext al proyecto (o espacio de nombres de Kubernetes). Para habilitar Astra Control Center para proteger su aplicación y mover la aplicación a otro clúster o proyecto en OpenShift, debe agregar directivas que permitan que la aplicación se ejecute como cualquier UID. Por ejemplo, los siguientes comandos de la CLI de OpenShift otorgan las directivas adecuadas a una aplicación de WordPress.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

Pasos

1. Seleccione **aplicaciones** y, a continuación, seleccione el nombre de una aplicación.
2. Seleccione **Protección de datos**.
3. Si desea restaurar desde una instantánea, mantenga seleccionado el icono **instantáneas**. De lo contrario, seleccione el icono **copias de seguridad** para restaurar desde una copia de seguridad.
4. En el menú Opciones de la columna **acciones** de la instantánea o copia de seguridad desde la que desea restaurar, seleccione **Restaurar aplicación**.
5. **Detalles de la restauración**: Especifique los detalles de la aplicación restaurada. De forma predeterminada, se muestran el clúster y el espacio de nombres actuales. Deje estos valores intactos para restaurar una aplicación in situ, que revierte la aplicación a una versión anterior de sí misma. Cambie estos valores si desea restaurar a un clúster o espacio de nombres diferentes.
   - Introduzca un nombre y un espacio de nombres para la aplicación.
   - Seleccione el clúster de destino de la aplicación.
   - Seleccione **Revisión**.

   > Si se restaura en un espacio de nombres que se eliminó previamente, se crea un espacio de nombres nuevo con el mismo nombre como parte del proceso de restauración. Cualquier usuario que tenga derechos para administrar aplicaciones en el espacio de nombres previamente eliminado debe restaurar manualmente los derechos en el espacio de nombres recién creado.

6. **Resumen de restauración**: Revise los detalles sobre la acción de restauración, escriba "restore" y seleccione **Restaurar**.

Resultado

Astra Control Center restaura la aplicación en función de la información proporcionada. Si restauró la aplicación en un lugar, el contenido de cualquier volumen persistente existente se reemplaza por el contenido de los volúmenes persistentes de la aplicación restaurada.

Después de una operación de protección de datos (clonado, backup, restauración) y un posterior cambio de tamaño de volumen persistente, se producen retrasos de hasta veinte minutos antes de que se muestre el nuevo tamaño del volumen en la interfaz de usuario web. La operación de protección de datos se realiza correctamente en cuestión de minutos, y se puede utilizar el software de gestión para el back-end de almacenamiento para confirmar el cambio de tamaño del volumen.

Clone y migre aplicaciones

Clone una aplicación existente para crear una aplicación duplicada en el mismo clúster de Kubernetes o en otro clúster. Cuando Astra Control Center clona una aplicación, crea un clon de la configuración de la aplicación y del almacenamiento persistente.

El clonado puede ayudarle si necesita mover aplicaciones y almacenamiento de un clúster de Kubernetes a otro. Por ejemplo, es posible que desee mover cargas de trabajo mediante una canalización de CI/CD y entre espacios de nombres Kubernetes. Puede utilizar la interfaz de usuario de Astra o. "La API de control Astra" para clonar y migrar aplicaciones.

Lo que necesitará

Para clonar aplicaciones en un clúster diferente, necesita un bloque predeterminado. Cuando se agrega su primer bloque, se convierte en el bloque predeterminado.

Acerca de esta tarea

- Si se implementa una aplicación con un StorageClass configurado explícitamente y se necesita clonar la aplicación, el clúster de destino debe tener el StorageClass especificado originalmente. Se producirá un error al clonar una aplicación con un tipo de almacenamiento establecido explícitamente en un clúster que no tenga el mismo tipo de almacenamiento.
- Si clona una instancia de Jenkins CI que ha puesto en marcha un operador, debe restaurar manualmente los datos persistentes. Esta es una limitación del modelo de puesta en marcha de la aplicación.
- Los bloques de S3 de Astra Control Center no informan sobre la capacidad disponible. Antes de realizar una copia de seguridad o clonar aplicaciones gestionadas por Astra Control Center, compruebe la información de los bloques en el sistema de gestión ONTAP o StorageGRID.
- Durante una copia de seguridad de la aplicación o una restauración de la aplicación, puede especificar un ID de bloque. Sin embargo, en una operación de clonado de aplicaciones, siempre se utiliza el bloque predeterminado que se ha definido. No existe ninguna opción para cambiar bloques para un clon. Si desea controlar qué segmento se utiliza, puede hacer lo mismo "cambiar el valor predeterminado del segmento" o haga un "Backup" seguido de un "restaurar" por separado.
- Cualquier usuario miembro con restricciones de espacio de nombres por nombre/ID de espacio de nombres o por etiquetas de espacio de nombres puede clonar o restaurar una aplicación en un nuevo espacio de nombres en el mismo clúster o en cualquier otro clúster de la cuenta de su organización. Sin embargo, el mismo usuario no puede acceder a la aplicación clonada o restaurada en el nuevo espacio de nombres. Después de crear un espacio de nombres nuevo mediante una operación de clonado o restauración, el propietario/administrador de la cuenta puede editar las restricciones de la cuenta de usuario miembro y actualizar las restricciones de roles para que el usuario afectado conceda acceso al nuevo espacio de nombres.

Consideraciones sobre OpenShift

- Si clona una aplicación entre clústeres, los clústeres de origen y destino deben ser la misma distribución de OpenShift. Por ejemplo, si clona una aplicación de un clúster de OpenShift 4.7, utilice un clúster de destino que también sea OpenShift 4.7.

- Al crear un proyecto para alojar una aplicación en un clúster de OpenShift, se asigna un UID de SecurityContext al proyecto (o espacio de nombres de Kubernetes). Para habilitar Astra Control Center para proteger su aplicación y mover la aplicación a otro clúster o proyecto en OpenShift, debe agregar directivas que permitan que la aplicación se ejecute como cualquier UID. Por ejemplo, los siguientes comandos de la CLI de OpenShift otorgan las directivas adecuadas a una aplicación de WordPress.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

Pasos

1. Seleccione **aplicaciones**.
2. Debe realizar una de las siguientes acciones:
   - Seleccione el menú Opciones de la columna **acciones** de la aplicación deseada.
   - Seleccione el nombre de la aplicación deseada y seleccione la lista desplegable de estado en la parte superior derecha de la página.
3. Seleccione **Clonar**.
4. **Detalles del clon**: Especifique los detalles del clon:
   - Introduzca un nombre.
   - Introduzca un espacio de nombres para el clon.
   - Elija un clúster de destino para el clon.
   - Elija si desea crear el clon a partir de una snapshot o un backup existente. Si no selecciona esta opción, Astra Control Center crea el clon a partir del estado actual de la aplicación.
5. **Fuente**: Si decide clonar desde una instantánea o copia de seguridad existente, elija la instantánea o copia de seguridad que desea utilizar.
6. Seleccione **Revisión**.
7. **Resumen de clones**: Revise los detalles sobre el clon y seleccione **clon**.

Resultado

Astra Control Center clona esa aplicación basándose en la información que nos ha proporcionado. La operación de clonado se realiza correctamente cuando el nuevo clon de la aplicación está en `Available` en la página **aplicaciones**.

> Después de una operación de protección de datos (clonado, backup, restauración) y un cambio de tamaño posterior de volumen persistente, se demora hasta veinte minutos antes de que se muestre el tamaño del nuevo volumen en la interfaz de usuario. La operación de protección de datos se realiza correctamente en cuestión de minutos, y se puede utilizar el software de gestión para el back-end de almacenamiento para confirmar el cambio de tamaño del volumen.

Gestione los enlaces de ejecución de aplicaciones

Un enlace de ejecución es una secuencia de comandos personalizada que se puede ejecutar antes o después de una instantánea de una aplicación administrada. Por ejemplo, si tiene una aplicación de base de datos, puede utilizar los enlaces de ejecución para pausar todas las transacciones de la base de datos antes de realizar una

instantánea y reanudar las transacciones una vez finalizada la instantánea. De este modo se garantiza la creación de instantáneas coherentes con la aplicación.

Ganchos de ejecución predeterminados y expresiones regulares

Para algunas aplicaciones, Astra Control incluye los enlaces de ejecución predeterminados, proporcionados por NetApp, que gestionan las operaciones de congelación y descongelación antes y después de las copias Snapshot. Astra Control utiliza expresiones regulares para relacionar la imagen de un contenedor de una aplicación con estas aplicaciones:

- MariaDB
  - Expresión regular coincidente: \Bmariadb\b
- MySQL
  - Expresión regular coincidente: \Bmysql\b
- PostgreSQL
  - Expresión regular coincidente: \Bpostgresql\b

Si hay algún dato, los enlaces de ejecución predeterminados proporcionados por NetApp para esa aplicación aparecen en la lista de enlaces de ejecución activos y dichos enlaces se ejecutan automáticamente cuando se hacen snapshots de esa aplicación. Si una de sus aplicaciones personalizadas tiene un nombre de imagen similar que se produce para coincidir con una de las expresiones regulares (y no desea utilizar los ganchos de ejecución predeterminados), puede cambiar el nombre de la imagen, o bien, desactive el enlace de ejecución predeterminado para esa aplicación y utilice un gancho personalizado en su lugar.

No puede eliminar ni modificar los enlaces de ejecución predeterminados.

Notas importantes sobre los enlaces de ejecución personalizados

Tenga en cuenta lo siguiente al planificar enlaces de ejecución para sus aplicaciones.

- Astra Control requiere que los enlaces de ejecución se escriban en el formato de secuencias de comandos de shell ejecutables.
- El tamaño del script está limitado a 128 KB.
- Astra Control utiliza la configuración del enlace de ejecución y cualquier criterio coincidente para determinar qué ganchos son aplicables a una instantánea.
- Todos los fallos del enlace de ejecución son errores de software; otros enlaces y la instantánea siguen intentándose incluso si falla un gancho. Sin embargo, cuando falla un gancho, se registra un suceso de advertencia en el registro de eventos de la página **Activity**.
- Para crear, editar o eliminar enlaces de ejecución, debe ser un usuario con permisos de propietario, administrador o miembro.
- Si un enlace de ejecución tarda más de 25 minutos en ejecutarse, el enlace fallará, creando una entrada de registro de eventos con un código de retorno de "N/A". Se agotará el tiempo de espera de todas las instantáneas afectadas y se marcarán como errores, con una entrada de registro de eventos resultante que tenga en cuenta el tiempo de espera.

Puesto que los enlaces de ejecución a menudo reducen o desactivan por completo la funcionalidad de la aplicación con la que se ejecutan, siempre debe intentar minimizar el tiempo que tardan los enlaces de ejecución personalizados.

Cuando se ejecuta una instantánea, los eventos de enlace de ejecución tienen lugar en el siguiente orden:

1. Todos los enlaces de ejecución presnapshot predeterminados que proporcione NetApp se ejecutan en los contenedores adecuados.
2. Todos los enlaces de ejecución de presnapshot personalizados aplicables se ejecutan en los contenedores adecuados. Puede crear y ejecutar tantos enlaces presnapshot personalizados como necesite, pero el orden de ejecución de estos enlaces antes de que la instantánea no esté garantizada ni sea configurable.
3. La copia de Snapshot se realiza.
4. Todos los enlaces de ejecución post-snapshot personalizados aplicables se ejecutan en los contenedores adecuados. Puede crear y ejecutar tantos enlaces post-snapshot personalizados como necesite, pero el orden de ejecución de estos enlaces después de que la instantánea no esté garantizada ni sea configurable.
5. Todos los enlaces de ejecución post-snapshot predeterminados que proporcione NetApp se ejecutan en los contenedores adecuados.

> Siempre debe probar sus secuencias de comandos de ejecución de enlace antes de habilitarlas en un entorno de producción. Puede utilizar el comando 'kubectl exec' para probar cómodamente los scripts. Después de habilitar los enlaces de ejecución en un entorno de producción, pruebe las instantáneas resultantes para asegurarse de que son coherentes. Para ello, puede clonar la aplicación en un espacio de nombres temporal, restaurar la instantánea y, a continuación, probar la aplicación.

Ver los enlaces de ejecución existentes

Puede ver los enlaces de ejecución predeterminados de una aplicación ya existentes o proporcionados por NetApp.

Pasos

1. Vaya a **aplicaciones** y seleccione el nombre de una aplicación administrada.
2. Seleccione la ficha **ganchos de ejecución**.

   Puede ver todos los enlaces de ejecución habilitados o desactivados en la lista resultante. Puede ver el estado, el origen y el momento en que se ejecuta un gancho (instantánea previa o posterior). Para ver los registros de eventos que rodean los enlaces de ejecución, vaya a la página **actividad** en el área de navegación del lado izquierdo.

Cree un enlace de ejecución personalizado

Puede crear un enlace de ejecución personalizado para una aplicación. Consulte "Ejemplos de gancho de ejecución" para ejemplos de gancho. Necesita tener permisos de propietario, administrador o miembro para crear enlaces de ejecución.

> Cuando cree un script de shell personalizado para utilizarlo como un enlace de ejecución, recuerde especificar el shell adecuado al principio del archivo, a menos que esté ejecutando comandos linux o proporcionando la ruta completa a un ejecutable.

Pasos

1. Seleccione **aplicaciones** y, a continuación, seleccione el nombre de una aplicación administrada.

2. Seleccione la ficha **ganchos de ejecución**.
3. Seleccione **Añadir un nuevo gancho**.
4. En el área **Detalles del gancho**, dependiendo de cuándo se debe ejecutar el gancho, elija **Pre-Snapshot** o **Post-Snapshot**.
5. Introduzca un nombre único para el gancho.
6. (Opcional) Introduzca cualquier argumento para pasar al gancho durante la ejecución, pulsando la tecla Intro después de cada argumento que introduzca para grabar cada uno.
7. En el área **Imágenes de contenedor**, si el gancho debe funcionar con todas las imágenes de contenedor contenidas en la aplicación, active la casilla de verificación **aplicar a todas las imágenes de contenedor**. Si en su lugar el gancho sólo debe actuar en una o más imágenes contenedoras especificadas, introduzca los nombres de imagen contenedora en el campo **nombres de imagen contenedora para que coincidan**.
8. En el área **Script**, siga uno de estos procedimientos:
   - Cargue un script personalizado.
     i. Seleccione la opción **cargar archivo**.
     ii. Navegue hasta un archivo y cárguelo.
     iii. Asigne al script un nombre único.
     iv. (Opcional) Introduzca cualquier nota que los otros administradores deben conocer sobre el script.
   - Pegar en un script personalizado desde el portapapeles.
     i. Seleccione la opción **Pegar del portapapeles**.
     ii. Seleccione el campo de texto y pegue el texto del script en el campo.
     iii. Asigne al script un nombre único.
     iv. (Opcional) Introduzca cualquier nota que los otros administradores deben conocer sobre el script.
9. Seleccione **Agregar gancho**.

Desactivar un gancho de ejecución

Puede desactivar un gancho de ejecución si desea impedir temporalmente que se ejecute antes o después de una instantánea de una aplicación. Necesita tener permisos de propietario, administrador o miembro para desactivar los enlaces de ejecución.

Pasos

1. Seleccione **aplicaciones** y, a continuación, seleccione el nombre de una aplicación administrada.
2. Seleccione la ficha **ganchos de ejecución**.
3. Seleccione el menú Opciones de la columna **acciones** para el gancho que desea desactivar.
4. Seleccione **Desactivar**.

Eliminar un gancho de ejecución

Puede eliminar un enlace de ejecución por completo si ya no lo necesita. Necesita tener permisos de propietario, administrador o miembro para eliminar los enlaces de ejecución.

Pasos

1. Seleccione **aplicaciones** y, a continuación, seleccione el nombre de una aplicación administrada.

2. Seleccione la ficha **ganchos de ejecución**.
3. Seleccione el menú Opciones de la columna **acciones** para el gancho que desea eliminar.
4. Seleccione **Eliminar**.

Ejemplos de gancho de ejecución

Utilice los siguientes ejemplos para obtener una idea de cómo estructurar los enlaces de ejecución. Puede utilizar estos enlaces como plantillas o como scripts de prueba.

Ejemplo de éxito simple

Este es un ejemplo de un simple enlace que se realiza correctamente y escribe un mensaje en la salida estándar y en un error estándar.

```
#!/bin/sh

# success_sample.sh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
```

```
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running success_sample.sh"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

Ejemplo de éxito simple (versión de bash)

Este es un ejemplo de un simple enlace que funciona y escribe un mensaje en la salida estándar y en un error estándar, escrito para bash.

```
#!/bin/bash

# success_sample.bash
#
# A simple noop success hook script for testing purposes.
#
# args: None


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
```

```
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running success_sample.bash"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

Ejemplo sencillo de éxito (versión zsh)

Este es un ejemplo de un simple enlace que se realiza correctamente y escribe un mensaje en la salida estándar y en un error estándar, escrito para el shell Z.

```
#!/bin/zsh

# success_sample.zsh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
```

```
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.zsh"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

Éxito con argumentos ejemplo

En el siguiente ejemplo se muestra cómo se pueden utilizar args en un gancho.

```
#!/bin/sh

# success_sample_args.sh
#
# A simple success hook script with args for testing purposes.
#
# args: Up to two optional args that are echoed to stdout
```

```
#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running success_sample_args.sh"


# collect args
arg1=$1
arg2=$2

# output args and arg count to stdout
info "number of args: $#"
info "arg1 ${arg1}"
info "arg2 ${arg2}"

# exit with 0 to indicate success
```

```
info "exit 0"
exit 0
```

Ejemplo de gancho de instantánea previa/posinstantánea

En el siguiente ejemplo se muestra cómo se puede utilizar el mismo script tanto para una instantánea previa como para un enlace posterior a una instantánea.

```
#!/bin/sh

# success_sample_pre_post.sh
#
# A simple success hook script example with an arg for testing purposes
# to demonstrate how the same script can be used for both a prehook and
posthook
#
# args: [pre|post]

# unique error codes for every error case
ebase=100
eusage=$((ebase+1))
ebadstage=$((ebase+2))
epre=$((ebase+3))
epost=$((ebase+4))


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
```

```
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# Would run prehook steps here
#
prehook() {
    info "Running noop prehook"
    return 0
}

#
# Would run posthook steps here
#
posthook() {
    info "Running noop posthook"
    return 0
}


#
# main
#

# check arg
stage=$1
if [ -z "${stage}" ]; then
    echo "Usage: $0 <pre|post>"
    exit ${eusage}
fi

if [ "${stage}" != "pre" ] && [ "${stage}" != "post" ]; then
    echo "Invalid arg: ${stage}"
    exit ${ebadstage}
fi

# log something to stdout
info "running success_sample_pre_post.sh"

if [ "${stage}" = "pre" ]; then
```

```
    prehook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during prehook"
    fi
fi

if [ "${stage}" = "post" ]; then
    posthook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during posthook"
    fi
fi

exit ${rc}
```

Ejemplo de fallo

En el siguiente ejemplo se muestra cómo puede controlar los fallos en un gancho.

```
#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
```

```
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}
```

Ejemplo de fallo detallado

En el ejemplo siguiente se muestra cómo puede controlar los errores en un enlace, con un registro más detallado.

```
#!/bin/sh

# failure_sample_verbose.sh
#
# A simple failure hook script with args for testing purposes.
#
# args: [The number of lines to output to stdout]


#
# Writes the given message to standard output
#
```

```
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running failure_sample_verbose.sh"


# output arg value to stdout
linecount=$1
info "line count ${linecount}"

# write out a line to stdout based on line count arg
i=1
while [ "$i" -le ${linecount} ]; do
    info "This is line ${i} from failure_sample_verbose.sh"
    i=$(( i + 1 ))
done

error "exiting with error code 8"
exit 8
```

Fallo con un ejemplo de código de salida

En el siguiente ejemplo se muestra un error de enlace con un código de salida.

```
#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#
```

```
# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}
```

Ejemplo de éxito tras fallo

El siguiente ejemplo muestra un gancho que falla la primera vez que se ejecuta, pero que tiene éxito después de la segunda ejecución.

```
#!/bin/sh

# failure_then_success_sample.sh
#
# A hook script that fails on initial run but succeeds on second run for
testing purposes.
#
# Helpful for testing retry logic for post hooks.
#
# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}
```

```
#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running failure_success sample.sh"


if [ -e /tmp/hook-test.junk ] ; then
    info "File does exist.  Removing /tmp/hook-test.junk"
    rm /tmp/hook-test.junk
    info "Second run so returning exit code 0"
    exit 0
else
    info "File does not exist.  Creating /tmp/hook-test.junk"
    echo "test" > /tmp/hook-test.junk
    error "Failed first run, returning exit code 5"
    exit 5
fi
```