Configure Astra Control Center

Astra Control Center

NetApp
April 25, 2024

Tabla de contenidos

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:

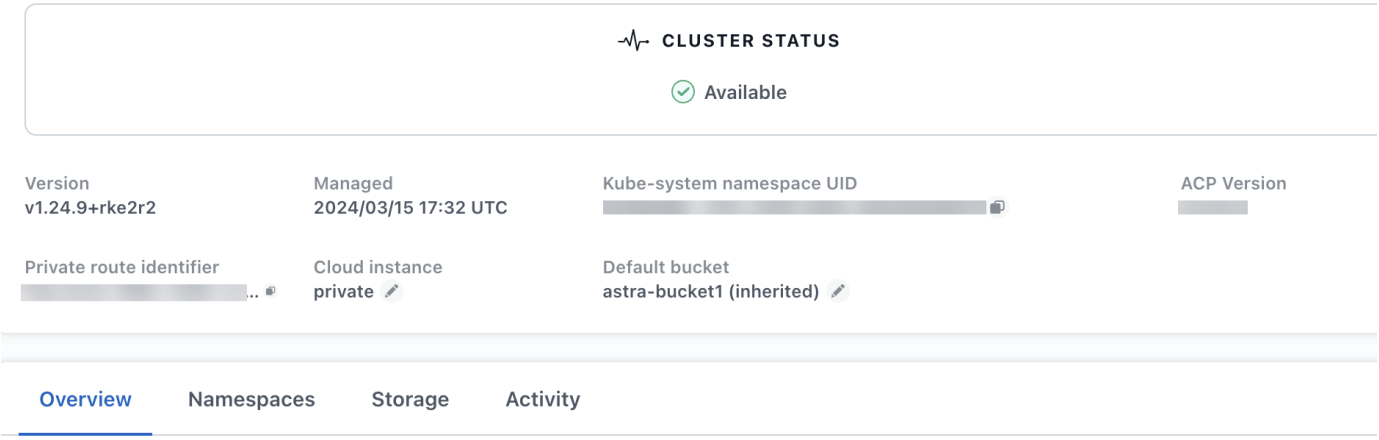
```
./tridentctl uninstall -n trident
```

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>
```

```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>
```

```
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
```

```yaml
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".