Configure o Astra Control Center

Astra Control Center

NetApp
August 11, 2025

Índice

Configure o Astra Control Center

Adicione uma licença para o Astra Control Center

Quando você instala o Astra Control Center, uma licença de avaliação incorporada já está instalada. Se você estiver avaliando o Astra Control Center, ignore esta etapa.

Você pode adicionar uma nova licença usando a IU do Astra Control ou "API Astra Control"o .

As licenças do Astra Control Center medem recursos de CPU usando unidades de CPU Kubernetes e contam os recursos de CPU atribuídos aos nós de trabalho de todos os clusters gerenciados do Kubernetes. As licenças são baseadas no uso do vCPU. Para obter mais informações sobre como as licenças são calculadas, "Licenciamento"consulte .

> Se a instalação aumentar para exceder o número licenciado de unidades de CPU, o Astra Control Center impedirá que você gerencie novas aplicações. É apresentado um alerta quando a capacidade é ultrapassada.

> Para atualizar uma avaliação existente ou uma licença completa, "Atualizar uma licença existente"consulte .

Antes de começar

- Acesso a uma instância recém-instalada do Astra Control Center.
- Permissões de função de administrador.
- A "Ficheiro de licença do NetApp" (NLF).

Passos

1. Faça login na IU do Astra Control Center.
2. Selecione **conta** > **Licença**.
3. Selecione **Adicionar licença**.
4. Navegue até o arquivo de licença (NLF) que você baixou.
5. Selecione **Adicionar licença**.

A página **Account** > **License** exibe as informações da licença, data de validade, número de série da licença, ID da conta e unidades CPU usadas.

> Se você tiver uma licença de avaliação e não estiver enviando dados para o AutoSupport, lembre-se de armazenar o ID da conta para evitar a perda de dados em caso de falha do Centro de Controle Astra.

Habilite o Astra Control Provisioner

O Astra Trident versões 23,10 e posteriores incluem a opção de usar o Astra Control Provisioner, que permite que usuários licenciados do Astra Control acessem o recurso avançado de provisionamento de storage. O Astra Control Provisioner fornece essa funcionalidade estendida, além da funcionalidade padrão baseada em CSI Astra Trident.

Nas próximas atualizações do Astra Control, o parceiro Astra Control substituirá o Astra Trident como provisionador de storage e orquestrador e será obrigatório para uso do Astra Control. Por causa disso, é altamente recomendável que os usuários do Astra Control ativem o Astra Control Provisioner. O Astra Trident continuará a ser de código aberto e será lançado, mantido, suportado e atualizado com o novo CSI e outros recursos do NetApp.

Sobre esta tarefa

Você deve seguir este procedimento se você for um usuário licenciado do Astra Control Center e estiver procurando usar a funcionalidade Astra Control Provisioner. Você também deve seguir este procedimento se você for um usuário do Astra Trident e quiser usar a funcionalidade adicional que o Astra Control Provisioner fornece sem usar também o Astra Control.

Para cada caso, a funcionalidade de provisionador não é habilitada por padrão no Astra Trident 24,02 e deve estar habilitada.

Antes de começar

Se você estiver habilitando o Astra Control Provisioner, faça o seguinte primeiro:

Astra Control visioners usuários com o Astra Control Center

- **Obter uma licença do Astra Control Center**: Você precisará de um "Licença do Astra Control Center" para habilitar o Astra Control Provisioner e acessar a funcionalidade que ele oferece.
- **Instalar ou atualizar para o Astra Control Center 23,10 ou posterior**: Você precisará da versão mais recente do Astra Control Center (24,02) se estiver planejando usar a funcionalidade mais recente do Astra Control Provisioner (24,02) com o Astra Control.
- **Confirme que seu cluster tem uma arquitetura de sistema AMD64**: A imagem Astra Control Provisioner é fornecida em arquiteturas de CPU AMD64 e ARM64, mas apenas AMD64 é compatível com o Astra Control Center.
- **Obtenha uma conta do Serviço Astra Control para acesso ao Registro**: Se você pretende usar o Registro Astra Control em vez do site de suporte da NetApp para fazer o download da imagem do programa Astra Control, preencha o Registro para um "Conta do Astra Control Service". após concluir e enviar o formulário e criar uma conta do BlueXP , você receberá um e-mail de boas-vindas do Serviço Astra Control.
- **Se você tiver o Astra Trident instalado, confirme que sua versão está dentro de uma janela de quatro versões**: Você pode fazer uma atualização direta para o Astra Trident 24,02 com o Astra Control Provisioner se o seu Astra Trident estiver dentro de uma janela de quatro versões da versão 24,02. Por exemplo, você pode fazer o upgrade diretamente do Astra Trident 23,04 para o 24,02.

Apenas usuários do Astra Control Provisioner

- **Obter uma licença do Astra Control Center**: Você precisará de um "Licença do Astra Control Center" para habilitar o Astra Control Provisioner e acessar a funcionalidade que ele oferece.
- **Se você tiver o Astra Trident instalado, confirme que sua versão está dentro de uma janela de quatro versões**: Você pode fazer uma atualização direta para o Astra Trident 24,02 com o Astra Control Provisioner se o seu Astra Trident estiver dentro de uma janela de quatro versões da versão 24,02. Por exemplo, você pode fazer o upgrade diretamente do Astra Trident 23,04 para o 24,02.
- **Obtenha uma conta do Astra Control Service para acesso ao Registro**: Você precisará de acesso ao Registro para baixar imagens do Astra Control Provisioner. Para começar, preencha o Registro para um "Conta do Astra Control Service". depois de preencher e enviar o formulário e criar uma conta do BlueXP , você receberá um e-mail de boas-vindas do Serviço Astra Control.

(Passo 1) Obtenha a imagem Astra Control Provisioner

Os usuários do Astra Control Center podem obter a imagem do Astra Control Provisioner usando o método do Registro Astra Control ou do site de suporte da NetApp. Os usuários do Astra Trident que desejam usar o Astra Control Provisioner sem o Astra Control devem usar o método de Registro.

Registro de imagem Astra Control

Você pode usar Podman em vez de Docker para os comandos neste procedimento. Se você estiver usando um ambiente Windows, o PowerShell é recomendado.

1. Acesse o Registro de imagem do NetApp Astra Control:
   a. Faça logon na IU da Web do Astra Control Service e selecione o ícone de figura no canto superior direito da página.
   b. Selecione **Acesso à API**.
   c. Anote o seu ID de conta.
   d. Na mesma página, selecione **Generate API token** e copie a cadeia de token da API para a área de transferência e salve-a no seu editor.
   e. Faça login no Registro Astra Control usando seu método preferido:

```
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. (Apenas registos personalizados) siga estes passos para mover a imagem para o seu registo personalizado. Se você não estiver usando um Registro, siga as etapas do operador Trident no "próxima seção".
   a. Extraia a imagem Astra Control Provisioner do Registro:

      A imagem puxada não suportará múltiplas plataformas e só suportará a mesma plataforma que o host que puxou a imagem, como o Linux AMD64.

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

   Exemplo:

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

   a. Marque a imagem:

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

b. Envie a imagem para o seu registo personalizado:

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

Você pode usar o Crane copy como alternativa para executar esses comandos do Docker: `crane copy cr.astra.netapp.io/astra/trident-acp:24.02.0 <my_custom_registry>/trident-acp:24.02.0`

Site de suporte da NetApp

1. Faça o download do pacote Astra Control Provisioner (`trident-acp-[version].tar`) no "Página de downloads do Astra Control Center".
2. (Recomendado, mas opcional) Faça o download do pacote de certificados e assinaturas para o Centro de Controle Astra (astra-control-center-certs-[version].tar.gz) para verificar a assinatura do pacote 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. Carregue a imagem do Astra Control Provisioner:

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

Resposta:

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

4. Marque a imagem:

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

5. Envie a imagem para o seu registo personalizado:

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

(Etapa 2) ative o Astra Control Provisioner no Astra Trident

Determine se o método de instalação original usou um "Operador (manualmente ou com Helm) ou tridentctl" e conclua as etapas apropriadas de acordo com o método original.

Operador do Astra Trident

1. "Baixe o instalador do Astra Trident e extraia-o.".
2. Siga estas etapas se você ainda não tiver instalado o Astra Trident ou se tiver removido o operador da sua implantação original do Astra Trident:
   a. Crie o CRD:

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

   b. Crie o namespace Trident (`kubectl create namespace trident`) ou confirme se o namespace Trident ainda existe (`kubectl get all -n trident`). Se o namespace tiver sido removido, crie-o novamente.
3. Atualize o Astra Trident para 24.02.0:

> Para clusters que executam o Kubernetes 1,24 ou anterior, `bundle_pre_1_25.yaml` use o . Para clusters que executam o Kubernetes 1,25 ou posterior, `bundle_post_1_25.yaml` use o .

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

4. Verifique se o Astra Trident está em execução:

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

   Resposta:

```
NAME      AGE
trident   21m
```

5. se você tem um Registro que usa segredos, crie um segredo para usar para puxar a imagem Astra Control Provisioner:

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

6. Edite o TridentOrchestrator CR e faça as seguintes edições:

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

a. Defina um local de Registro personalizado para a imagem Astra Trident ou extraia-a do Registro Astra Control (`tridentImage: <my_custom_registry>/trident:24.02.0`ou `tridentImage: netapp/trident:24.02.0`).

b. Ative o Astra Control Provisioner (`enableACP: true`).

c. Defina o local de Registro personalizado para a imagem Astra Control Provisioner ou extraia-a do Registro Astra Control (`acpImage: <my_custom_registry>/trident-acp:24.02.0`ou `acpImage: cr.astra.netapp.io/astra/trident-acp:24.02.0`).

d. Se tiver estabelecido a imagem puxa segredos anteriormente neste procedimento, pode defini-los aqui (`imagePullSecrets: - <secret_name>`). Use o mesmo nome secreto que você estabeleceu nas etapas 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. Salve e saia do arquivo. O processo de implantação começará automaticamente.
8. Verifique se o operador, a implantação e as replicasets são criados.

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

Deve haver apenas **uma instância** do operador em um cluster do Kubernetes. Não crie várias implantações do operador Astra Trident.

9. Verifique se o `trident-acp` contentor está em execução e se `acpVersion` está `24.02.0` com um status de `Installed`:

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

Resposta:

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

tridentctl

1. "Baixe o instalador do Astra Trident e extraia-o.".
2. "Se você tiver um Astra Trident existente, desinstale-o do cluster que o hospeda".
3. Instalar o Astra Trident com a previsão de controle Astra ativada (`--enable-acp=true`):

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

4. Confirme se o Astra Control Provisioner foi ativado:

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

Resposta:

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

Leme

1. Se tiver o Astra Trident 23.07.1 ou anterior instalado, "desinstalar" o operador e outros componentes.
2. Se o cluster do Kubernetes estiver executando o 1,24 ou anterior, exclua a psp:

```
kubectl delete psp tridentoperatorpod
```

3. Adicione o repositório Astra Trident Helm:

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

4. Atualize o gráfico Helm:

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

Resposta:

```
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. Liste as imagens:

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

Resposta:

```
| 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. Certifique-se de que o Trident-Operator 24.02.0 está disponível:

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

Resposta:

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

7. Utilize `helm install` e execute uma das seguintes opções que incluem estas definições:
   - Um nome para o local de implantação
   - A versão Astra Trident
   - O nome da imagem Astra Control Provisioner
   - A bandeira para habilitar o provisionador
   - (Opcional) Um caminho de Registro local. Se você estiver usando um Registro local, o "Imagens de Trident" pode estar localizado em um Registro ou Registros diferentes, mas todas as imagens CSI devem estar localizadas no mesmo Registro.
   - O namespace Trident

Opções

- Imagens sem registo

```
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
```

- Imagens em um ou mais 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
```

Você pode usar `helm list` para revisar detalhes de instalação, como nome, namespace, gráfico, status, versão do aplicativo e número de revisão.

Se você tiver algum problema na implantação do Trident usando o Helm, execute este comando para desinstalar completamente o Astra Trident:

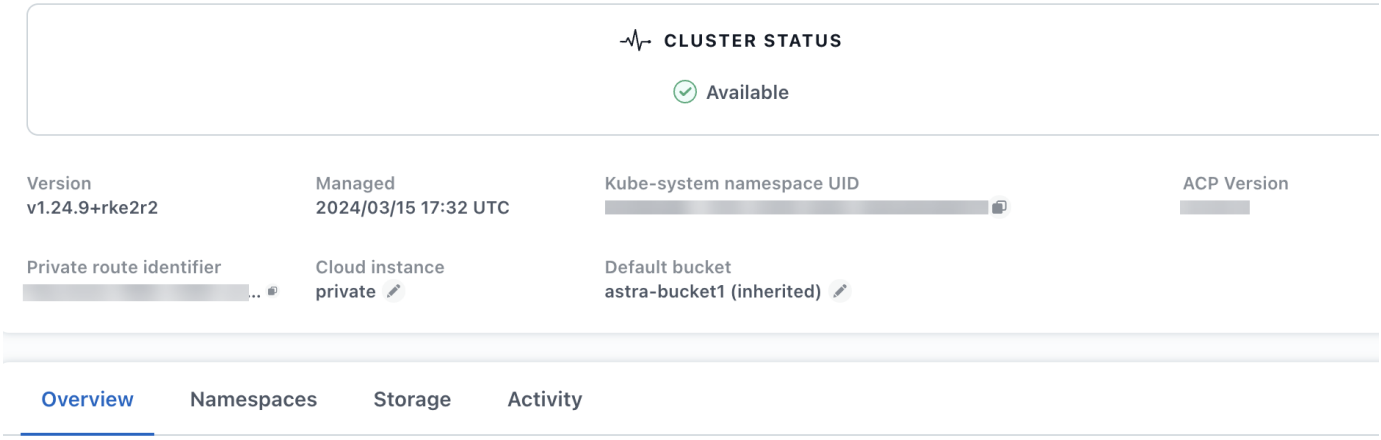
```
./tridentctl uninstall -n trident
```

Não "Remova completamente CRDS Astra Trident" como parte da sua desinstalação antes de tentar ativar o Astra Control Provisioner novamente.

Resultado

A funcionalidade Astra Control Provisioner está ativada e você pode usar todos os recursos disponíveis para a versão em execução.

(Somente para usuários do Astra Control Center) após a instalação do Astra Control Provisioner, o cluster que hospeda o provisionador na IU do Astra Control Center mostrará um `ACP version` número de versão instalado em vez `Trident version` de campo e atual.



Para mais informações

- "O Astra Trident atualiza a documentação"

Prepare seu ambiente para gerenciamento de clusters com o Astra Control

Você deve garantir que as seguintes condições de pré-requisito sejam atendidas antes de adicionar um cluster. Você também deve executar verificações de qualificação para garantir que seu cluster esteja pronto para ser adicionado ao Astra Control Center e criar funções de cluster kubeconfig conforme necessário.

O Astra Control permite adicionar clusters gerenciados por recursos personalizados (CR) ou kubeconfig, dependendo do seu ambiente e preferências.

Antes de começar

- **Atenda aos pré-requisitos ambientais**: Seu ambiente atende "requisitos do ambiente operacional" ao Astra Control Center.
- **Configurar nós de trabalho**: Certifique-se de que você "configure os nós de trabalho" esteja em seu cluster com os drivers de armazenamento apropriados para que os pods possam interagir com o armazenamento de back-end.
- **Habilitar restrições PSA**: Se o cluster tiver a aplicação de admissão de segurança do pod ativada, o que é padrão para clusters do Kubernetes 1,25 e posteriores, você precisa ativar restrições de PSA nesses namespaces:

- `netapp-acc-operator` namespace:

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

- `netapp monitoring` namespace:

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

- * Credenciais ONTAP*: Você precisa de credenciais ONTAP e um superusuário e ID de usuário definidos no sistema ONTAP de backup para fazer backup e restaurar aplicativos com o Astra Control Center.

  Execute os seguintes comandos na linha de comando 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 cluster gerenciados por kubeconfig**: Esses requisitos são específicos para clusters de aplicativos gerenciados pelo kubeconfig.
  - **Tornar o kubeconfig acessível**: Você tem acesso ao "cluster predefinido kubeconfig" "você configurou durante a instalação"that .
  - **Considerações de autoridade de certificação**: Se você estiver adicionando o cluster usando um arquivo kubeconfig que faça referência a uma autoridade de certificação privada (CA), adicione a seguinte linha à `cluster` seção do arquivo kubeconfig. Isso permite que o Astra Control adicione o cluster:

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

  - **Somente Rancher**: Ao gerenciar clusters de aplicativos em um ambiente Rancher, modifique o contexto padrão do cluster de aplicativos no arquivo kubeconfig fornecido pelo Rancher para usar um contexto de plano de controle em vez do contexto do servidor da API Rancher. Isso reduz a carga no servidor de API Rancher e melhora o desempenho.
- **Requisitos da previsão do Astra Control**: Você deve ter um programa de controle Astra Control configurado corretamente, incluindo seus componentes do Astra Trident, para gerenciar clusters.
  - * Rever os requisitos de ambiente do Astra Trident*: Antes de instalar ou atualizar o Astra Control Provisioner, revise o "interfaces suportadas, backends e configurações de host".
  - **Ativar a funcionalidade do programa Astra Control**: É altamente recomendável instalar o Astra Trident 23,10 ou posterior e ativar "Funcionalidade de storage avançada do Astra Control Provisioner"o . Nos próximos lançamentos, o Astra Control não será compatível com o Astra Trident se o programa Astra Control também não estiver habilitado.

- **Configurar um back-end de armazenamento**: Pelo menos um back-end de armazenamento deve estar "Configurado no Astra Trident" no cluster.
- **Configurar uma classe de armazenamento**: Pelo menos uma classe de armazenamento deve estar "Configurado no Astra Trident" no cluster. Se uma classe de armazenamento padrão estiver configurada, certifique-se de que é a classe de armazenamento **only** que tem a anotação padrão.
- **Configure um controlador de snapshot de volume e instale uma classe de snapshot de volume**: "Instale um controlador instantâneo de volume" Para que os snapshots possam ser criados no Astra Control. "Criar" Pelo menos um `VolumeSnapshotClass` usando Astra Trident.

Execute verificações de qualificação

Execute as seguintes verificações de qualificação para garantir que o cluster esteja pronto para ser adicionado ao Astra Control Center.

Passos

1. Determine a versão do Astra Trident que você está executando:

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

   Se o Astra Trident existir, você verá uma saída semelhante à seguinte:

   ```
   NAME      VERSION
   trident   24.02.0
   ```

   Se o Astra Trident não existir, você verá uma saída semelhante à seguinte:

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

2. Execute um dos seguintes procedimentos:
   - Se você estiver executando o Astra Trident 23,01 ou anterior, use-os "instruções" para atualizar para uma versão mais recente do Astra Trident antes de atualizar para o Astra Control Provisioner. Você pode "faça uma atualização direta" usar o Astra Control Provisioner 24,02 se o seu Astra Trident estiver dentro de uma janela de quatro versões da versão 24,02. Por exemplo, você pode fazer o upgrade diretamente do Astra Trident 23,04 para o Astra Control Provisioner 24,02.
   - Se você estiver executando o Astra Trident 23,10 ou posterior, verifique se o Astra Control Provisioner foi "ativado". O Astra Control Provisioner não funcionará com versões do Astra Control Center anteriores a 23,10. "Atualize seu Astra Control Provisioner" Para que ele tenha a mesma versão do Astra Control Center que você está atualizando para acessar as funcionalidades mais recentes.
3. Verifique se todos os pods (`trident-acp`incluindo) estão em execução:

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

4. Determine se as classes de storage estão usando os drivers Astra Trident compatíveis. O nome do provisionador deve ser `csi.trident.netapp.io`. Veja o exemplo a seguir:

```
kubectl get sc
```

Resposta da amostra:

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

Crie uma função de cluster kubeconfig

Para clusters gerenciados usando o kubeconfig, você pode, opcionalmente, criar uma função de administrador de permissão limitada ou expandida para o Astra Control Center. Este não é um procedimento necessário para a configuração do Astra Control Center, uma vez que já configurou um kubeconfig como parte do "processo de instalação".

Este procedimento ajuda você a criar um kubeconfig separado se qualquer um dos seguintes cenários se aplicar ao seu ambiente:

- Você deseja limitar as permissões do Astra Control nos clusters que ele gerencia
- Você usa vários contextos e não pode usar o kubeconfig padrão do Astra Control configurado durante a instalação ou uma função limitada com um único contexto não funcionará em seu ambiente

Antes de começar

Certifique-se de que tem o seguinte para o cluster que pretende gerir antes de concluir as etapas do procedimento:

- kubectl v1,23 ou posterior instalado
- Acesso kubectl ao cluster que você pretende adicionar e gerenciar com o Astra Control Center

> Para esse procedimento, você não precisa de acesso kubectl ao cluster que está executando o Astra Control Center.

- Um kubeconfig ativo para o cluster que pretende gerir com direitos de administrador de cluster para o contexto ativo

Passos

1. Criar uma conta de serviço:
   a. Crie um arquivo de conta de serviço `astracontrol-service-account.yaml` chamado .

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

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

b. Aplique a conta de serviço:

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

2. Crie uma das seguintes funções de cluster com permissões suficientes para que um cluster seja gerenciado pelo Astra Control:

Função limitada do cluster

Essa função contém as permissões mínimas necessárias para que um cluster seja gerenciado pelo Astra Control:

a. Crie um `ClusterRole` arquivo chamado, por exemplo `astra-admin-account.yaml`, .

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

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

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

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

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

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

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

```
  - imagestreams/layers
  - imagestreamtags
  - imagetags
  verbs:
  - update
```

b. (Somente para clusters OpenShift) Append o seguinte no final `astra-admin-account.yaml` do arquivo:

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

c. Aplique a função de cluster:

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

Função expandida do cluster

Essa função contém permissões expandidas para um cluster a ser gerenciado pelo Astra Control. Você pode usar essa função se você usar vários contextos e não puder usar o kubeconfig padrão do Astra Control configurado durante a instalação ou uma função limitada com um único contexto não funcionará em seu ambiente:

> As etapas a seguir `ClusterRole` são um exemplo geral do Kubernetes. Consulte a documentação da distribuição do Kubernetes para obter instruções específicas para o seu ambiente.

a. Crie um `ClusterRole` arquivo chamado, por exemplo `astra-admin-account.yaml`, .

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

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

b. Aplique a função de cluster:

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

3. Crie a vinculação de função de cluster para a função de cluster à conta de serviço:

a. Crie um `ClusterRoleBinding` arquivo chamado `astracontrol-clusterrolebinding.yaml`.

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

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

b. Aplicar a vinculação de funções do cluster:

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

4. Crie e aplique o segredo do token:
   a. Crie um arquivo secreto de token `secret-astracontrol-service-account.yaml` chamado .

```
<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 o segredo do token:

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

5. Adicione o segredo do token à conta de serviço adicionando seu nome ao `secrets` array (a última linha no exemplo a seguir):

```
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. Liste os segredos da conta de serviço, substituindo `<context>` pelo contexto correto para sua instalação:

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

O final da saída deve ser semelhante ao seguinte:

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

Os índices para cada elemento no `secrets` array começam com 0. No exemplo acima, o índice para `astracontrol-service-account-dockercfg-48xhx` seria 0 e o índice para `secret-astracontrol-service-account` seria 1. Na sua saída, anote o número do índice para o segredo da conta de serviço. Você precisará desse número de índice na próxima etapa.

7. Gere o kubeconfigo da seguinte forma:
   a. Crie um `create-kubeconfig.sh` arquivo.
   b. Substitua `TOKEN_INDEX` no início do script a seguir pelo valor correto.

```
<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. Forneça os comandos para aplicá-los ao cluster do Kubernetes.

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

8. (Opcional) Renomear o kubeconfig para um nome significativo para o cluster.

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

(Prévia técnica) Instalar o Astra Connector para clusters gerenciados

Os clusters gerenciados pelo Astra Control Center usam o Astra Connector para permitir a comunicação entre o cluster gerenciado e o Astra Control Center. É necessário instalar o Astra Connector em todos os clusters que você deseja gerenciar.

Instale o conetor Astra

Você instala o Astra Connector usando comandos Kubernetes e arquivos de recursos personalizados (CR).

Sobre esta tarefa

- Ao executar essas etapas, execute esses comandos no cluster que deseja gerenciar com o Astra Control.
- Se você estiver usando um host de bastião, emita esses comandos a partir da linha de comando do host de bastião.

Antes de começar

- Você precisa ter acesso ao cluster que deseja gerenciar com o Astra Control.
- Você precisa de permissões de administrador do Kubernetes para instalar o operador Astra Connector no cluster.

Se o cluster estiver configurado com imposição de admissão de segurança de pod, que é o padrão para clusters Kubernetes 1,25 e posteriores, será necessário habilitar restrições PSA nos namespaces apropriados. "Prepare seu ambiente para gerenciamento de clusters com o Astra Control"Consulte para obter instruções.

Passos

1. Instale o operador do conetor Astra no cluster que você deseja gerenciar com o Astra Control. Quando você executa esse comando, o namespace `astra-connector-operator` é criado e a configuração é aplicada ao namespace:

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

2. Verifique se o operador está instalado e pronto:

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

3. Obtenha um token de API do Astra Control. Consulte o "Documentação do Astra Automation" para obter instruções.
4. Crie um segredo usando o token. Substitua o <API_TOKEN> pelo token recebido do Astra Control:

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

5. Crie um segredo do Docker para usar para puxar a imagem do conetor Astra. Substitua os valores entre parêntesis> por informações do seu ambiente:

Você pode encontrar o <ASTRA_CONTROL_ACCOUNT_ID> na IU da Web do Astra Control. Na IU da Web, selecione o ícone de figura no canto superior direito da página e selecione **Acesso à 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. Crie o arquivo CR do Astra Connector e nomeie-o `astra-connector-cr.yaml`. Atualize os valores entre parêntesis> para corresponder ao seu ambiente Astra Control e à configuração de cluster:
   - <ASTRA_CONTROL_ACCOUNT_ID>: Obtido na IU da Web do Astra Control durante a etapa anterior.
   - <CLUSTER_NAME>: O nome que esse cluster deve ser atribuído no Astra Control.

- <ASTRA_CONTROL_URL>: O URL da IU da Web do Astra Control. Por exemplo:

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

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

7. Depois de preencher o `astra-connector-cr.yaml` ficheiro com os valores corretos, aplique o CR:

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

8. Verifique se o conetor Astra está totalmente implantado:

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

9. Verifique se o cluster está registrado no Astra Control:

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

Você deve ver saída semelhante ao seguinte:

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

10. Verifique se o cluster aparece na lista de clusters gerenciados na página **clusters** da IU da Web Astra Control.

Adicione um cluster

Para começar a gerenciar suas aplicações, adicione um cluster do Kubernetes e gerencie-o como um recurso de computação. Você precisa adicionar um cluster para Astra Control Center para descobrir suas aplicações Kubernetes.

Recomendamos que o Astra Control Center gerencie o cluster em que ele é implantado primeiro antes de adicionar outros clusters ao Astra Control Center para gerenciar. Ter o cluster inicial sob gerenciamento é necessário enviar dados do Kubemetrics e dados associados ao cluster para métricas e solução de problemas.

Antes de começar

- Antes de adicionar um cluster, revise e execute o "tarefas pré-requisitos"necessário .
- Se você estiver usando um driver SAN ONTAP, verifique se o multipath está ativado em todos os clusters Kubernetes.

Passos

1. Navegue pelo menu Dashboard ou clusters:
   - Em **Dashboard** no Resumo de recursos, selecione **Add** no painel clusters.
   - Na área de navegação à esquerda, selecione **clusters** e, em seguida, selecione **Adicionar cluster** na página clusters.
2. Na janela **Add Cluster** que se abre, carregue um `kubeconfig.yaml` ficheiro ou cole o conteúdo de um `kubeconfig.yaml` ficheiro.

   O `kubeconfig.yaml` arquivo deve incluir **somente a credencial de cluster para um cluster**.

   Se você criar seu próprio `kubeconfig` arquivo, você deve definir apenas **um** elemento de contexto nele. "Documentação do Kubernetes"Consulte para obter informações sobre a criação `kubeconfig` de ficheiros. Se você criou um kubeconfig para uma função de cluster limitada usando "este processo"o , certifique-se de carregar ou colar esse kubeconfig nesta etapa.

3. Forneça um nome de credencial. Por padrão, o nome da credencial é preenchido automaticamente como o nome do cluster.
4. Selecione **seguinte**.
5. Selecione a classe de armazenamento padrão a ser usada para este cluster Kubernetes e selecione **Next**.

Você deve selecionar uma classe de storage configurada no Astra Control Provisioner com o suporte do ONTAP Storage.

6. Revise as informações e, se tudo estiver bem, selecione **Adicionar**.

Resultado

O cluster entra no estado **Descobrindo** e depois muda para **saudável**. Agora você está gerenciando o cluster com Astra Control Center.

Depois de adicionar um cluster a ser gerenciado no Astra Control Center, talvez demore alguns minutos para implantar o operador de monitoramento. Até então, o ícone de notificação fica vermelho e Registra um evento **Falha na verificação do status do agente de monitoramento**. Você pode ignorar isso, porque o problema resolve quando o Astra Control Center obtém o status correto. Se o problema não resolver em alguns minutos, vá para o cluster e execute `oc get pods -n netapp-monitoring` como ponto de partida. Você precisará examinar os logs do operador de monitoramento para depurar o problema.

Habilitar a autenticação em um back-end de storage do ONTAP

O Astra Control Center oferece dois modos de autenticação de um back-end do ONTAP:

- **Autenticação baseada em credenciais**: O nome de usuário e senha para um usuário do ONTAP com as permissões necessárias. Você deve usar uma função de login de segurança pré-definida, como admin ou vsadmin para garantir a máxima compatibilidade com as versões do ONTAP.
- **Autenticação baseada em certificado**: O Astra Control Center também pode se comunicar com um cluster ONTAP usando um certificado instalado no back-end. Você deve usar o certificado de cliente, a chave e o certificado de CA confiável, se usado (recomendado).

Você pode atualizar posteriormente os backends existentes para passar de um tipo de autenticação para outro método. Apenas um método de autenticação é suportado de cada vez.

Ative a autenticação baseada em credenciais

O Astra Control Center requer as credenciais para um cluster com escopo `admin` para se comunicar com o back-end do ONTAP. Você deve usar funções padrão e predefinidas, `admin` como . Isso garante compatibilidade direta com futuras versões do ONTAP que podem expor APIs de recursos a serem usadas por futuras versões do Astra Control Center.

Uma função de login de segurança personalizada pode ser criada e usada com o Astra Control Center, mas não é recomendada.

Uma definição de backend de exemplo se parece com esta:

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

A definição de back-end é o único lugar onde as credenciais são armazenadas em texto simples. A criação ou atualização de um backend é a única etapa que requer conhecimento das credenciais. Como tal, é uma operação somente de administração, realizada pelo Kubernetes ou pelo administrador de storage.

Ativar autenticação baseada em certificado

O Centro de Controle Astra pode usar certificados para se comunicar com backends ONTAP novos e existentes. Você deve inserir as seguintes informações na definição de back-end.

- `clientCertificate`: Certificado do cliente.
- `clientPrivateKey`: Chave privada associada.
- `trustedCACertificate`: Certificado de CA confiável. Se estiver usando uma CA confiável, esse parâmetro deve ser fornecido. Isso pode ser ignorado se nenhuma CA confiável for usada.

Você pode usar um dos seguintes tipos de certificados:

- Certificado auto-assinado
- Certificado de terceiros

Ative a autenticação com um certificado autoassinado

Um fluxo de trabalho típico envolve as etapas a seguir.

Passos

1. Gerar um certificado e chave de cliente. Ao gerar, defina o Nome Comum (CN) para o usuário ONTAP para autenticar 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 o certificado de cliente de tipo `client-ca` e chave no cluster do 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 se a função de login de segurança do ONTAP suporta o método de autenticação 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. Teste a autenticação usando o certificado gerado. Substitua o ONTAP Management LIF> e o <vserver name> pelo IP de LIF de gerenciamento e nome da SVM. Você deve garantir que o LIF tenha sua política de serviço definida como `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. Usando os valores obtidos na etapa anterior, adicione o back-end de storage na IU do Astra Control Center.

Ative a autenticação com um certificado de terceiros

Se você tiver um certificado de terceiros, poderá configurar a autenticação baseada em certificado com estas etapas.

Passos

1. Gerar a chave privada e 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. Passe o CSR para a CA do Windows (CA de terceiros) e emita o certificado assinado.
3. Baixe o certificado assinado e nomeie-o como "ONTAP_signed_cert.crt"
4. Exporte o certificado raiz da CA do Windows (CA de terceiros).
5. Nomeie este arquivo `ca_root.crt`

   Agora você tem os seguintes três arquivos:

   - **Chave privada**: `ontap_signed_request.key` (Esta é a chave correspondente para o certificado do

servidor no ONTAP. É necessário ao instalar o certificado do servidor.)

- **Certificado assinado**: `ontap_signed_cert.crt` (Isso também é chamado de *certificado do servidor* no ONTAP.)
- **Certificado CA raiz**: (Também é chamado de *certificado CA* *`ca_root.crt`* *Server-CA* no ONTAP.)

6. Instale estes certificados no ONTAP. Gerar, instalar `server` e `server-ca` certificados no 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. Crie o certificado de cliente para o mesmo host para comunicação sem senha. O Centro de Controle Astra usa esse processo para se comunicar com o ONTAP.
8. Gerar e instalar os certificados de cliente no 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. Adicione o back-end de storage à IU do Astra Control Center e forneça os seguintes valores:
   - **Certificado do cliente**: ONTAP_test_client.pem
   - **Chave privada**: ONTAP_test_client.key
   - **Certificado de CA confiável**: ONTAP_signed_cert.crt

Adicionar um back-end de storage

Depois de configurar as credenciais ou as informações de autenticação de certificado, você poderá adicionar um back-end de storage do ONTAP existente ao Astra Control Center para gerenciar seus recursos.

O gerenciamento de clusters de storage no Astra Control como um back-end de storage permite que você tenha vínculos entre volumes persistentes (PVS) e o back-end de storage, bem como métricas de storage adicionais.

Adicionar e gerenciar back-ends de storage do ONTAP no Astra Control Center é opcional ao usar a

tecnologia NetApp SnapMirror se você tiver ativado o Astra Control Provisioner.

Passos

1. No Painel na área de navegação à esquerda, selecione **backends**.
2. Selecione **Adicionar**.
3. Na seção usar existente da página Adicionar storage backend, selecione **ONTAP**.
4. Selecione uma das seguintes opções:
   - **Use as credenciais de administrador**: Insira o endereço IP e as credenciais de administrador de gerenciamento de cluster do ONTAP. As credenciais devem ser credenciais de todo o cluster.

     O usuário cujas credenciais você inserir aqui deve ter o `ontapi` método de acesso de login de usuário habilitado no Gerenciador de sistema do ONTAP no cluster do ONTAP. Se você planeja usar a replicação do SnapMirror, aplique credenciais de usuário com a função "admin", que tem os métodos de acesso `ontapi` e `http`, nos clusters ONTAP de origem e destino. "Gerenciar contas de usuário na documentação do ONTAP"Consulte para obter mais informações.

   - **Use um certificado**: Carregue o arquivo de certificado `.pem`, o arquivo de chave de certificado `.key` e, opcionalmente, o arquivo de autoridade de certificação.
5. Selecione **seguinte**.
6. Confirme os detalhes do backend e selecione **Manage**.

Resultado

O backend aparece no `online` estado da lista com informações de resumo.

Talvez seja necessário atualizar a página para que o backend apareça.

Adicione um balde

Você pode adicionar um bucket usando a IU do Astra Control ou "API Astra Control"o . Adicionar fornecedores de bucket do armazenamento de objetos é essencial para fazer backup das aplicações e do storage persistente ou clonar aplicações entre clusters. O Astra Control armazena os backups ou clones nos buckets do armazenamento de objetos que você define.

Você não precisa de um bucket no Astra Control se estiver clonando a configuração da aplicação e o storage persistente para o mesmo cluster. A funcionalidade de instantâneos de aplicações não requer um intervalo.

Antes de começar

- Garanta que você tenha um bucket acessível a partir dos clusters gerenciados pelo Astra Control Center.
- Certifique-se de que tem credenciais para o bucket.
- Certifique-se de que o balde é um dos seguintes tipos:
  - NetApp ONTAP S3
  - NetApp StorageGRID S3
  - Microsoft Azure

- Genérico S3

> A Amazon Web Services (AWS) e o Google Cloud Platform (GCP) usam o tipo de bucket Generic S3.

> Embora o Astra Control Center ofereça suporte ao Amazon S3 como um provedor de bucket do Generic S3, o Astra Control Center pode não oferecer suporte a todos os fornecedores de armazenamento de objetos que claim o suporte ao S3 da Amazon.

Passos

1. Na área de navegação à esquerda, selecione **Buckets**.
2. Selecione **Adicionar**.
3. Selecione o tipo de balde.

> Quando você adiciona um bucket, selecione o provedor de bucket correto e forneça as credenciais certas para esse provedor. Por exemplo, a IU aceita o NetApp ONTAP S3 como o tipo e aceita credenciais StorageGRID; no entanto, isso fará com que todos os backups e restaurações futuros de aplicativos que usam esse bucket falhem.

4. Insira um nome de bucket existente e uma descrição opcional.

> O nome e a descrição do bucket aparecem como um local de backup que você pode escolher mais tarde ao criar um backup. O nome também aparece durante a configuração da política de proteção.

5. Introduza o nome ou endereço IP do endpoint S3.
6. Em **Selecionar credenciais**, escolha a guia **Adicionar** ou **usar existente**.
   - Se você escolheu **Add**:
     i. Insira um nome para a credencial que a distingue de outras credenciais no Astra Control.
     ii. Insira a ID de acesso e a chave secreta colando o conteúdo da área de transferência.
   - Se você escolheu **Use existing**:
     i. Selecione as credenciais existentes que você deseja usar com o bucket.
7. `Add`Selecione .

> Quando você adiciona um balde, o Astra Control marca um balde com o indicador de balde padrão. O primeiro bucket que você criar se torna o bucket padrão. À medida que você adiciona buckets, você pode decidir mais tarde "defina outro intervalo padrão".

Informações sobre direitos autorais

Informações sobre marcas comerciais