Comece agora

Astra Control Center

NetApp
August 11, 2025

Índice

Comece agora

Saiba mais sobre o Astra Control

O Astra Control é uma solução de gerenciamento de ciclo de vida de dados de aplicações Kubernetes que simplifica as operações de aplicações com estado monitorado. Proteja, faça backup, replique e migre workloads do Kubernetes com facilidade e crie clones de aplicações em funcionamento instantaneamente.

Caraterísticas

O Astra Control oferece funcionalidades essenciais para o gerenciamento do ciclo de vida dos dados da aplicação Kubernetes:

- Gerencie automaticamente o storage persistente
- Crie backups e snapshots sob demanda com reconhecimento de aplicações
- Automatizar operações de backup e snapshot orientadas por políticas
- Migrar aplicações e dados entre clusters do Kubernetes
- Replique aplicações para um sistema remoto usando a tecnologia NetApp SnapMirror (Astra Control Center)
- Clonar aplicações da preparação para a produção
- Visualize a integridade e o status de proteção da aplicação
- Trabalhe com uma IU da Web ou uma API para implementar seus fluxos de trabalho de backup e migração

Modelos de implantação

O Astra Control está disponível em dois modelos de implantação:

- **Astra Control Service**: Um serviço gerenciado pelo NetApp que fornece gerenciamento de dados com reconhecimento de aplicações dos clusters do Kubernetes em vários ambientes de fornecedores de nuvem e clusters do Kubernetes autogerenciados.
- **Astra Control Center**: Software autogerenciado que oferece gerenciamento de dados com reconhecimento de aplicações dos clusters Kubernetes executados em seu ambiente local. O Astra Control Center também pode ser instalado em vários ambientes de fornecedor de nuvem com um back-end de storage da NetApp Cloud Volumes ONTAP.

	Astra Control Service	Astra Control Center
Como é oferecido?	Como um serviço de nuvem totalmente gerenciado da NetApp	Como software que você pode baixar, instalar e gerenciar
Onde está hospedado?	Em uma nuvem pública de escolha da NetApp	No seu próprio cluster Kubernetes
Como é atualizado?	Gerenciado por NetApp	Você gerencia quaisquer atualizações

	Astra Control Service	Astra Control Center
Quais são as distribuições compatíveis do Kubernetes?	• **Provedores de nuvem** ◦ Amazon Web Services ▪ Amazon Elastic Kubernetes Service (EKS) ◦ Google Cloud ▪ Google Kubernetes Engine (GKE) ◦ Microsoft Azure ▪ Serviço Kubernetes do Azure (AKS) • **Clusters autogeridos** ◦ Kubernetes (upstream) ◦ Rancher Kubernetes Engine (RKE) ◦ Red Hat OpenShift Container Platform • **Clusters locais** ◦ Red Hat OpenShift Container Platform no local	• Serviço Kubernetes do Azure no Azure Stack HCI • Google Anthos • Kubernetes (upstream) • Rancher Kubernetes Engine (RKE) • Red Hat OpenShift Container Platform

	Astra Control Service	Astra Control Center
Quais são os backends de armazenamento suportados?	• **Provedores de nuvem** ◦ Amazon Web Services ▪ Amazon EBS ▪ Amazon FSX para NetApp ONTAP ▪ "Cloud Volumes ONTAP" ◦ Google Cloud ▪ Persistent Disk do Google ▪ NetApp Cloud Volumes Service ▪ "Cloud Volumes ONTAP" ◦ Microsoft Azure ▪ Discos gerenciados do Azure ▪ Azure NetApp Files ▪ "Cloud Volumes ONTAP" • **Clusters autogeridos** ◦ Amazon EBS ◦ Discos gerenciados do Azure ◦ Persistent Disk do Google ◦ "Cloud Volumes ONTAP" ◦ NetApp MetroCluster ◦ "Longhorn" • **Clusters locais** ◦ NetApp MetroCluster ◦ Sistemas NetApp ONTAP AFF e FAS ◦ NetApp ONTAP Select ◦ "Cloud Volumes ONTAP" ◦ "Longhorn"	• Sistemas NetApp ONTAP AFF e FAS • NetApp ONTAP Select • "Cloud Volumes ONTAP" • "Longhorn"

Como funciona o Astra Control Service

O Astra Control Service é um serviço de nuvem gerenciado pela NetApp que está sempre ativo e atualizado com as funcionalidades mais recentes. Ele utiliza vários componentes para habilitar o gerenciamento do ciclo de vida dos dados das aplicações.

Em um alto nível, o Astra Control Service funciona assim:

- Você começa a usar o Astra Control Service configurando seu fornecedor de nuvem e registrando-se em uma conta Astra.

    - Para clusters GKE, o Astra Control Service usa "NetApp Cloud Volumes Service para Google Cloud" ou discos persistentes do Google como back-end de storage para volumes persistentes.
    - Para clusters AKS, o Astra Control Service usa "Azure NetApp Files" ou discos gerenciados do Azure como o back-end de storage para seus volumes persistentes.
    - Para clusters do Amazon EKS, o Astra Control Service usa "Amazon Elastic Block Store" ou "Amazon FSX para NetApp ONTAP" como back-end de storage para volumes persistentes.
- Você adiciona sua primeira computação do Kubernetes ao Astra Control Service. Em seguida, o Astra Control Service faz o seguinte:
    - Cria um armazenamento de objetos na sua conta de fornecedor de nuvem, que é onde as cópias de backup são armazenadas.

      No Azure, o Astra Control Service também cria um grupo de recursos, uma conta de storage e chaves para o contêiner de Blob.

    - Cria uma nova função de administrador e conta de serviço do Kubernetes no cluster.
    - Usa essa nova função de administrador para instalar o link../conceitos/arquitetura no cluster e para criar uma ou mais classes de storage.
    - Se você usa uma oferta de storage de serviço de nuvem da NetApp como back-end de storage, o Astra Control Service usa o Astra Control Provisioner para provisionar volumes persistentes para suas aplicações. Se você usar os discos gerenciados do Amazon EBS ou Azure como back-end de armazenamento, precisará instalar um driver CSI específico do provedor. As instruções de instalação são fornecidas na "Configurar o Amazon Web Services" e "Configurar o Microsoft Azure com discos gerenciados do Azure".
- Neste ponto, você pode adicionar aplicativos ao cluster. Volumes persistentes serão provisionados na nova classe de armazenamento padrão.
- Depois, você usa o Astra Control Service para gerenciar essas aplicações e começar a criar snapshots, backups e clones.

O Plano Gratuito do Astra Control permite gerenciar até 10 namespaces em sua conta. Se você quiser gerenciar mais de 10, precisará configurar o faturamento atualizando do Plano Gratuito para o Plano Premium.

Como funciona o Astra Control Center

Astra Control Center é executado localmente em sua própria nuvem privada.

O Astra Control Center é compatível com clusters de Kubernetes com uma classe de storage configurada para Provisioner Astra Control com um back-end de storage da ONTAP.

Monitoramento e telemetria limitados (7 dias de métricas) estão disponíveis no Astra Control Center e também exportados para ferramentas de monitoramento nativas do Kubernetes (como Prometheus e Grafana) por meio de pontos finais de métricas abertas.

O Astra Control Center é totalmente integrado ao ecossistema de consultores digitais da AutoSupport e Active IQ (também conhecido como consultor digital) para fornecer aos usuários e ao suporte da NetApp informações de solução de problemas e uso.

Você pode experimentar o Astra Control Center usando uma licença de avaliação incorporada de 90 dias. Enquanto você está avaliando o Astra Control Center, pode obter suporte por meio de e-mail e opções da comunidade. Além disso, você tem acesso a artigos e documentação da base de conhecimento a partir do painel de suporte do produto.

Para instalar e usar o Astra Control Center, você precisará atender a determinados "requisitos".

Em um alto nível, o Astra Control Center funciona assim:

- Você instala o Astra Control Center em seu ambiente local. Saiba mais sobre como "Instale o Astra Control Center" .
- Você conclui algumas tarefas de configuração, como estas:
  - Configure o licenciamento.
  - Adicione o primeiro cluster.
  - Adicione o back-end de storage descoberto quando você adicionou o cluster.
  - Adicione um bucket do armazenamento de objetos que armazenará os backups do aplicativo.

Saiba mais sobre como "Configure o Astra Control Center" .

Você pode adicionar aplicativos ao cluster. Ou, se você já tiver algumas aplicações no cluster sendo gerenciado, poderá usar o Astra Control Center para gerenciá-las. Depois, use o Astra Control Center para criar snapshots, backups, clones e relacionamentos de replicação.

Para mais informações

- "Documentação do Astra Control Service"
- "Documentação do Astra Control Center"
- "Documentação do Astra Trident"
- "Documentação da API Astra Control"
- "Documentação do ONTAP"

Requisitos do Astra Control Center

Comece verificando a prontidão do seu ambiente operacional, clusters de aplicativos, aplicativos, licenças e navegador da Web. Garanta que seu ambiente atenda a esses requisitos para implantar e operar o Astra Control Center.

Ambientes de Kubernetes do cluster de host compatíveis

O Astra Control Center foi validado com os seguintes ambientes de host do Kubernetes:

Garantir que o ambiente do Kubernetes que você escolher hospedar o Astra Control Center atenda aos requisitos básicos de recursos descritos na documentação oficial do ambiente.

Distribuição do Kubernetes no cluster de host	Versões suportadas
Serviço Kubernetes do Azure no Azure Stack HCI	Azure Stack HCI 21H2 e 22H2 com AKS 1.24.11 a 1.26.6
Google Anthos	1,15 a 1,16 (Requisitos de entrada do Google Anthosconsulte)
Kubernetes (upstream)	1,27 a 1,29

Distribuição do Kubernetes no cluster de host	Versões suportadas
Rancher Kubernetes Engine (RKE)	RKE 1: Versões 1.24.17, 1.25.13, 1.26.8 com Rancher Manager 2.7.9 RKE 2: Versões 1.23.16 e 1.24.13 com Rancher Manager 2.6.13 RKE 2: Versões 1.24.17, 1.25.14, 1.26.9 com Rancher Manager 2.7.9
Red Hat OpenShift Container Platform	4,12 a 4,14

Requisitos de recursos de cluster de host

O Astra Control Center requer os seguintes recursos, além dos requisitos de recursos do ambiente:

> Esses requisitos presumem que o Astra Control Center é a única aplicação em execução no ambiente operacional. Se o ambiente estiver executando aplicativos adicionais, ajuste esses requisitos mínimos de acordo.

- * Extensões de CPU*: As CPUs em todos os nós do ambiente de hospedagem devem ter extensões AVX ativadas.
- **Worker Nodes**: Pelo menos 3 worker node total, com 4 núcleos de CPU e 12GB GB de RAM cada
- **Requisitos de cluster do VMware Tanzu Kubernetes Grid**: Ao hospedar o Astra Control Center em um cluster do VMware Tanzu Kubernetes Grid (TKG) ou Tanzu Kubernetes Grid Integrated Edition (TKGI), tenha em mente as seguintes considerações.
  - O token de arquivo de configuração padrão do VMware TKG e TKGI expira dez horas após a implantação. Se você usa produtos do portfólio Tanzu, precisará gerar um arquivo de configuração de cluster do Kubernetes da Tanzu com um token sem expiração para evitar problemas de conexão entre o Astra Control Center e os clusters de aplicativos gerenciados. Para obter instruções, visite "Documentação do produto do data center VMware NSX-T."
  - Use o `kubectl get nsxlbmonitors -A` comando para ver se você já tem um monitor de serviço configurado para aceitar o tráfego de entrada. Se existir um, não deve instalar o MetalLB, porque o monitor de serviço existente substituirá qualquer nova configuração do balanceador de carga.
  - Desative a aplicação da classe de armazenamento padrão TKG ou TKGI em qualquer cluster de aplicativos que seja gerenciado pelo Astra Control. Você pode fazer isso editando o `TanzuKubernetesCluster` recurso no cluster do namespace.
  - Esteja ciente dos requisitos específicos do Astra Control Provisioner quando você implantar o Astra Control Center em um ambiente TKG ou TKGI:
    - O cluster precisa dar suporte a workloads privilegiados.
    - A `--kubelet-dir` bandeira deve ser definida para a localização do diretório kubelet. Por padrão, isso é `/var/vcap/data/kubelet`.
    - Especificar a localização do kubelet usando `--kubelet-dir` é conhecido por funcionar para o Operador Trident, Helm e `tridentctl` implantações.

Requisitos de malha de serviço

É altamente recomendável instalar uma versão vanilla compatível da malha de serviço Istio no cluster de host Astra Control Center. "versões compatíveis"Consulte para obter as versões compatíveis do Istio. As versões com marca do serviço Mesh Istio, como OpenShift Service Mesh, não são validadas com o Astra Control Center.

Para integrar o Astra Control Center à malha de serviço Istio instalada no cluster de host, é necessário fazer a integração como parte de um Astra Control Center "instalação"e não independente desse processo.

Instalar e usar o Astra Control Center sem configurar uma malha de serviço no cluster de host tem implicações potencialmente sérias na segurança.

Astra Trident

Se você pretende usar o Astra Trident em vez do Astra Control Provisioner com este lançamento, o Astra Trident 23,04 e versões posteriores são compatíveis. O Astra Control Center exigirá Previsão do Astra Control em versões futuras.

Previsão do Astra Control

Para usar o recurso avançado de storage do Astra Control Provisioner, você deve instalar o Astra Trident 23,10 ou posterior e ativar "Funcionalidade do Astra Control Provisioner"o . Para usar a funcionalidade mais recente do Astra Control Provisioner, você precisará das versões mais recentes do Astra Trident e do Astra Control Center.

- **Versão mínima do Astra Control Provisioner para uso com o Astra Control Center**: Astra Control Provisioner 23,10 ou posterior instalado e configurado.

Configuração de ONTAP com Astra Trident

- **Storage class**: Configure pelo menos uma classe de armazenamento no cluster. Se uma classe de armazenamento padrão estiver configurada, verifique se ela é a única classe de armazenamento com a designação padrão.
- **Drivers de armazenamento e nós de trabalho**: Certifique-se de configurar os nós de trabalho no cluster com os drivers de armazenamento apropriados para que os pods possam interagir com o armazenamento de back-end. O Astra Control Center é compatível com os seguintes drivers ONTAP fornecidos pelo Astra Trident:
  - `ontap-nas`
  - `ontap-san`
  - `ontap-san-economy` (a replicação de aplicativos não está disponível com esse tipo de classe de storage)
  - `ontap-nas-economy` (snapshots e políticas de replicação de aplicativos não estão disponíveis com esse tipo de classe de storage)

Back-ends de armazenamento

Certifique-se de ter um back-end compatível com capacidade suficiente.

- **Capacidade de back-end de armazenamento necessária**: Pelo menos 500GB GB disponíveis
- **Backends compatíveis**: O Astra Control Center é compatível com os seguintes back-ends de storage:
  - NetApp ONTAP 9.9,1 ou sistemas AFF, FAS e ASA posteriores
  - NetApp ONTAP Select 9.9.1 ou posterior
  - NetApp Cloud Volumes ONTAP 9.9.1 ou posterior
  - (Para pré-visualização técnica do Astra Control Center) NetApp ONTAP 9.10,1 ou posterior para

operações de proteção de dados fornecidas como prévia técnica

- Longhorn 1.5.0 ou posterior
  - Requer a criação manual de um objeto VolumeSnapshotClass. Consulte o "Documentação de Longhorn" para obter instruções.
- NetApp MetroCluster
  - Os clusters do Kubernetes gerenciado precisam estar em uma configuração mais ampla.
- Back-ends de armazenamento disponíveis com provedores de nuvem compatíveis

Licenças ONTAP

Para usar o Astra Control Center, verifique se você tem as seguintes licenças do ONTAP, dependendo do que você precisa realizar:

- FlexClone
- SnapMirror: Opcional. Necessário apenas para replicação para sistemas remotos usando a tecnologia SnapMirror. Consulte a "Informações de licença do SnapMirror".
- Licença S3: Opcional. Necessário apenas para buckets do ONTAP S3

Para verificar se o sistema ONTAP tem as licenças necessárias, "Gerenciar licenças do ONTAP" consulte .

NetApp MetroCluster

Ao usar o NetApp MetroCluster como um back-end de storage, você precisa fazer o seguinte:

- Especifique um LIF de gerenciamento de SVM como uma opção de back-end no driver Astra Trident que você usa
- Certifique-se de que tem a licença ONTAP adequada

Para configurar o MetroCluster LIF, consulte estas opções e exemplos para cada driver:

- "SAN"
- "NAS"

Licença do Astra Control Center

O Astra Control Center requer uma licença do Astra Control Center. Quando você instala o Astra Control Center, uma licença de avaliação incorporada de 90 dias para 4.800 unidades CPU já está ativada. Se você precisar de mais capacidade ou termos de avaliação diferentes ou quiser atualizar para uma licença completa, você pode obter uma licença de avaliação diferente ou uma licença completa da NetApp. Você precisa de uma licença para proteger seus aplicativos e dados.

Você pode experimentar o Astra Control Center inscrevendo-se para uma avaliação gratuita. Você pode se inscrever registrando "aqui".

Para configurar a licença, "use uma licença de avaliação de 90 dias"consulte a .

Para saber mais sobre como as licenças funcionam, "Licenciamento"consulte .

Requisitos de rede

Configure seu ambiente operacional para garantir que o Astra Control Center possa se comunicar corretamente. São necessárias as seguintes configurações de rede:

- **Endereço FQDN**: Você deve ter um endereço FQDN para o Astra Control Center.
- **Acesso à internet**: Você deve determinar se tem acesso externo à internet. Se não o fizer, algumas funcionalidades poderão ser limitadas, como o envio de pacotes de suporte para o "Site de suporte da NetApp".
- **Acesso à porta**: O ambiente operacional que hospeda o Astra Control Center se comunica usando as seguintes portas TCP. Você deve garantir que essas portas sejam permitidas por meio de firewalls e configurar firewalls para permitir qualquer tráfego de saída HTTPS proveniente da rede Astra. Algumas portas exigem conectividade entre o ambiente que hospeda o Astra Control Center e cada cluster gerenciado (observado quando aplicável).

É possível implantar o Astra Control Center em um cluster de Kubernetes de duas stack e o Astra Control Center pode gerenciar aplicações e back-ends de storage configurados para operação de duas stack. Para obter mais informações sobre os requisitos de cluster de pilha dupla, consulte o "Documentação do Kubernetes".

Fonte	Destino	Porta	Protocolo	Finalidade
PC do cliente	Astra Control Center	443	HTTPS	UI / API Access - Certifique-se de que essa porta esteja aberta em ambas as direções entre o Astra Control Center e o sistema usado para acessar o Astra Control Center
Consumidor de métricas	Nó de trabalho do Astra Control Center	9090	HTTPS	Comunicação de dados de métricas - garanta que cada cluster gerenciado possa acessar essa porta no cluster que hospeda o Astra Control Center (comunicação bidirecional necessária)
Astra Control Center	Fornecedor de bucket de storage do Amazon S3	443	HTTPS	Comunicação de armazenamento Amazon S3
Astra Control Center	NetApp AutoSupport (https://support.netapp.com)	443	HTTPS	Comunicação NetApp AutoSupport

Fonte	Destino	Porta	Protocolo	Finalidade
Astra Control Center	Cluster gerenciado do Kubernetes	443/6443 **NOTA**: A porta que o cluster gerenciado usa pode variar dependendo do cluster. Consulte a documentação do fornecedor de software de cluster.	HTTPS	Comunicação com cluster gerenciado - garanta que essa porta esteja aberta de ambas as maneiras entre o cluster que hospeda o Astra Control Center e cada cluster gerenciado

Entrada para clusters do Kubernetes no local

Você pode escolher o tipo de entrada de rede que o Astra Control Center usa. Por padrão, o Astra Control Center implanta o gateway Astra Control Center (Service/traefik) como um recurso em todo o cluster. O Astra Control Center também é compatível com o uso de um balanceador de carga de serviço, se permitido no seu ambiente. Se você preferir usar um balanceador de carga de serviço e ainda não tiver um configurado, você pode usar o balanceador de carga MetalLB para atribuir automaticamente um endereço IP externo ao serviço. Na configuração do servidor DNS interno, você deve apontar o nome DNS escolhido para o Astra Control Center para o endereço IP com balanceamento de carga.

> O balanceador de carga deve usar um endereço IP localizado na mesma sub-rede que os endereços IP do nó de trabalho do Astra Control Center.

Para obter mais informações, "Configure a entrada para o balanceamento de carga"consulte .

Requisitos de entrada do Google Anthos

Ao hospedar o Astra Control Center em um cluster do Google Anthos, observe que o Google Anthos inclui o balanceador de carga MetalLB e o serviço de ingresso Istio por padrão, permitindo que você simplesmente use os recursos genéricos de entrada do Astra Control Center durante a instalação. "Documentação de instalação do Astra Control Center"Consulte para obter detalhes.

Navegadores da Web suportados

O Astra Control Center suporta versões recentes do Firefox, Safari e Chrome com uma resolução mínima de 1280 x 720.

Requisitos adicionais para clusters de aplicações

Tenha em mente esses requisitos se você planeja usar esses recursos do Astra Control Center:

- * Requisitos de cluster de aplicativos*: "Requisitos de gerenciamento de clusters"
  - **Requisitos de aplicação gerenciada**: "Requisitos de gerenciamento de aplicativos"
  - **Requisitos adicionais para replicação de aplicativos**: "Pré-requisitos de replicação"

O que vem a seguir

Veja a "início rápido" visão geral.

Início rápido para Astra Control Center

Aqui está uma visão geral das etapas necessárias para começar a usar o Astra Control Center. Os links em cada etapa levam você a uma página que fornece mais detalhes.

1 Analisar os requisitos do cluster do Kubernetes

Certifique-se de que seu ambiente atenda a esses requisitos:

Cluster do Kubernetes

- "Certifique-se de que o cluster de host atenda aos requisitos do ambiente operacional"
- "Configurar o ingresso para balanceamento de carga de clusters do Kubernetes no local"

Integração de armazenamento

- "Garanta que seu ambiente inclua o Astra Control Provisioner"
- "Habilite os recursos avançados de gerenciamento e provisionamento de storage do Astra Control Provisioner"
- "Preparar nós de trabalho de cluster"
- "Configurar backends de armazenamento"
- "Configurar classes de armazenamento"
- "Instale um controlador instantâneo de volume"
- "Crie uma classe de instantâneo de volume"

Credenciais ONTAP

- "Configurar credenciais do ONTAP"

2 Baixe e instale o Astra Control Center

Conclua estas tarefas de instalação:

- "Faça download do Centro de Controle Astra na página de downloads do site de suporte da NetApp"
- Obtenha o ficheiro de licença NetApp:
  - Se você estiver avaliando o Astra Control Center, uma licença de avaliação incorporada já estará incluída
  - "Se você já comprou o Astra Control Center, gere seu arquivo de licença"
- "Instale o Astra Control Center"
- "Execute etapas de configuração opcionais adicionais"

3 Conclua algumas tarefas de configuração inicial

Conclua algumas tarefas básicas para começar:

- "Adicione uma licença"

- "Prepare seu ambiente para o gerenciamento de clusters"
- "Adicione um cluster"
- "Adicionar um back-end de storage"
- "Adicione um balde"

4 **Use o Astra Control Center**

Depois de concluir a configuração do Astra Control Center, use a IU do Astra Control ou a "API Astra Control" para começar a gerenciar e proteger aplicações:

- "Gerenciar contas": Usuários, funções, LDAP, credenciais e muito mais.
- "Gerenciar notificações"
- "Gerir aplicações": Defina recursos para gerenciar.
- "Proteja aplicativos": Configurar políticas de proteção e replicar, clonar e migrar aplicativos.

Para mais informações

- "Use a API Astra Control"
- "Atualizar o Astra Control Center"
- "Obtenha ajuda com o Astra Control"

Visão geral da instalação

Escolha e conclua um dos seguintes procedimentos de instalação do Astra Control Center:

- "Instale o Astra Control Center usando o processo padrão"
- "(Se você usar o Red Hat OpenShift) instale o Astra Control Center usando o OpenShift OperatorHub"
- "Instalar o Astra Control Center com um back-end de storage do Cloud Volumes ONTAP"

Dependendo do seu ambiente, pode haver configuração adicional necessária após a instalação do Astra Control Center:

- "Configure o Astra Control Center após a instalação"

Instale o Astra Control Center usando o processo padrão

Para instalar o Astra Control Center, baixe as imagens de instalação e execute as seguintes etapas. Você pode usar este procedimento para instalar o Astra Control Center em ambientes conetados à Internet ou com conexão via rede.

Para uma demonstração do processo de instalação do Astra Control Center, "este vídeo" consulte .

Antes de começar

- * Atender pré-requisitos ambientais *"Antes de começar a instalação, prepare seu ambiente para a implantação do Astra Control Center": .

Implante o Astra Control Center em um domínio de terceiros ou local secundário. Isso é recomendado para replicação de aplicativos e recuperação de desastres aprimorada.

- * Garantir serviços saudáveis*: Verifique se todos os serviços de API estão em um estado saudável e disponíveis:

```
kubectl get apiservices
```

- **Certifique-se de que um FQDN roteável**: O FQDN Astra que você planeja usar pode ser roteado para o cluster. Isso significa que você tem uma entrada DNS no seu servidor DNS interno ou está usando uma rota URL principal que já está registrada.
- **Configure cert Manager**: Se um gerenciador de cert já existir no cluster, você precisará executar alguns "etapas de pré-requisito" para que o Astra Control Center não tente instalar seu próprio gerenciador de cert. Por padrão, o Astra Control Center instala seu próprio gerenciador de cert durante a instalação.
- **(somente driver SAN ONTAP) Ativar multipath**: Se você estiver usando um driver SAN ONTAP, verifique se o multipath está habilitado em todos os clusters Kubernetes.

Você também deve considerar o seguinte:

- **Tenha acesso ao Registro de imagens do NetApp Astra Control**:

  Você tem a opção de obter imagens de instalação e melhorias de funcionalidade para o Astra Control, como o Astra Control Provisioner, a partir do Registro de imagens do NetApp.

  a. Registre seu ID de conta Astra Control que você precisará fazer login no Registro.

     Você pode ver o ID da conta na IU da Web do Astra Control Service. Selecione o ícone de figura no canto superior direito da página, selecione **Acesso à API** e anote o ID da sua conta.

  b. 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.

  c. Faça login no Registro do Astra Control:

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

- **Instalar uma malha de serviço para comunicações seguras**: É altamente recomendável que os canais de comunicação do cluster de host Astra Control sejam protegidos usando um "malha de serviço suportada".

  A integração do Astra Control Center com uma malha de serviço só pode ser feita durante o Astra Control Center "instalação"e não independente desse processo. A alteração de um ambiente de malha para um ambiente sem malha não é suportada.

  Para uso em malha de serviço do Istio, você precisará fazer o seguinte:

  - Adicione um `istio-injection:enabled` etiquetaao namespace Astra antes de implantar o Astra Control Center.
  - Utilize o `Generic` definição de entrada e forneça uma entrada alternativa para balanceamento de

carga externoo .

- Para clusters do Red Hat OpenShift, você precisa definir `NetworkAttachmentDefinition` em todos os namespaces associados do Astra Control Center (`netapp-acc-operator` `netapp-acc`, , `netapp-monitoring` para clusters de aplicativos ou quaisquer namespaces personalizados que tenham sido substituídos).

```
cat <<EOF | oc -n netapp-acc-operator create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-acc create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-monitoring create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF
```

Passos

Para instalar o Astra Control Center, siga estas etapas:

- Faça download e extraia Astra Control Center
- Conclua as etapas adicionais se você usar um Registro local
- Configure namespace e segredo para Registros com requisitos de autenticação
- Instale o operador do Centro de Controle Astra
- Configurar o Astra Control Center
- Instalação completa do operador e do Centro de Controle Astra
- Verifique o status do sistema
- Configure a entrada para o balanceamento de carga
- Faça login na IU do Astra Control Center

> Não exclua o operador Astra Control Center (por exemplo, `kubectl delete -f astra_control_center_operator_deploy.yaml`) a qualquer momento durante a instalação ou operação do Astra Control Center para evitar a exclusão de pods.

Faça download e extraia Astra Control Center

Faça o download das imagens do Astra Control Center de um dos seguintes locais:

- **Registro de imagem do Serviço de Controle Astra**: Use esta opção se você não usar um Registro local com as imagens do Centro de Controle Astra ou se preferir esse método para o download do pacote no site de suporte da NetApp.
- **Site de suporte da NetApp**: Use essa opção se você usar um Registro local com as imagens do Centro de Controle Astra.

Registro de imagem Astra Control

1. Faça login no Astra Control Service.
2. No Dashboard, selecione **Deploy a self-managed instance of Astra Control**.
3. Siga as instruções para fazer login no Registro de imagens do Astra Control, extrair a imagem de instalação do Astra Control Center e extrair a imagem.

Site de suporte da NetApp

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

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

   ```
   openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub
   -signature certs/astra-control-center-[version].tar.gz.sig astra-
   control-center-[version].tar.gz
   ```

   A saída será `Verified OK` exibida após a verificação bem-sucedida.

3. Extraia as imagens do pacote Astra Control Center:

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

Conclua as etapas adicionais se você usar um Registro local

Se você está planejando enviar o pacote Astra Control Center para o seu Registro local, você precisa usar o plugin de linha de comando NetApp Astra kubectl.

Instale o plug-in NetApp Astra kubectl

Conclua estas etapas para instalar o plugin de linha de comando mais recente do NetApp Astra kubectl.

Antes de começar

O NetApp fornece binários de plug-in para diferentes arquiteturas de CPU e sistemas operacionais. Você precisa saber qual CPU e sistema operacional você tem antes de executar esta tarefa.

Se você já tiver o plugin instalado a partir de uma instalação anterior, "certifique-se de que tem a versão mais recente" antes de concluir estas etapas.

Passos

1. Liste os binários disponíveis do plug-in NetApp Astra kubectl:

   A biblioteca de plugins kubectl faz parte do pacote tar e é extraída para a pasta `kubectl-astra`.

   ```
   ls kubectl-astra/
   ```

2. Mova o arquivo necessário para o sistema operacional e a arquitetura da CPU para o caminho atual e renomeie-o para `kubectl-astra`:

   ```
   cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
   ```

Adicione as imagens ao seu registo

1. Se você estiver planejando enviar o pacote Astra Control Center para o Registro local, conclua a sequência de etapas apropriada para o mecanismo de contêiner:

Docker

a. Mude para o diretório raiz do tarball. Você deve ver o `acc.manifest.bundle.yaml` arquivo e estes diretórios:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Envie as imagens do pacote no diretório de imagens do Astra Control Center para o Registro local. Faça as seguintes substituições antes de executar o `push-images` comando:

- Substitua o <BUNDLE_FILE> pelo nome do arquivo do pacote Astra Control (`acc.manifest.bundle.yaml`).
- Substitua o <MY_FULL_REGISTRY_PATH> pela URL do repositório Docker; por exemplo"<a href="https://<docker-registry>"" class="bare">https://<docker-registry>", .
- Substitua o <MY_REGISTRY_USER> pelo nome de usuário.
- Substitua o <MY_REGISTRY_TOKEN> por um token autorizado para o Registro.

```
kubectl astra packages push-images -m <BUNDLE_FILE> -r
<MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
<MY_REGISTRY_TOKEN>
```

Podman

a. Mude para o diretório raiz do tarball. Você deve ver este arquivo e diretório:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Inicie sessão no seu registo:

```
podman login <YOUR_REGISTRY>
```

c. Prepare e execute um dos seguintes scripts personalizados para a versão do Podman que você usa. Substitua o <MY_FULL_REGISTRY_PATH> pela URL do seu repositório que inclui quaisquer subdiretórios.

```
<strong>Podman 4</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

O caminho da imagem que o script cria deve ser semelhante ao seguinte, dependendo da configuração do Registro:

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/24.02.0-69/image:version
```

2. Altere o diretório:

```
cd manifests
```

Configure namespace e segredo para Registros com requisitos de autenticação

1. Exporte o kubeconfig para o cluster de host Astra Control Center:

```
export KUBECONFIG=[file path]
```

Antes de concluir a instalação, certifique-se de que seu kubeconfig esteja apontando para o cluster onde você deseja instalar o Astra Control Center.

2. Se você usar um Registro que requer autenticação, você precisará fazer o seguinte:

   a. Crie o `netapp-acc-operator` namespace:

```
kubectl create ns netapp-acc-operator
```

   b. Crie um segredo para o `netapp-acc-operator` namespace. Adicione informações do Docker e execute o seguinte comando:

O marcador de posição `your_registry_path` deve corresponder à localização das imagens que carregou anteriormente (por exemplo, `[Registry_URL]/netapp/astra/astracc/24.02.0-69`).

```
kubectl create secret docker-registry astra-registry-cred -n netapp-acc-
operator --docker-server=cr.astra.netapp.io --docker
-username=[astra_account_id] --docker-password=[astra_api_token]
```

+

```
kubectl create secret docker-registry astra-registry-cred -n netapp-acc-
operator --docker-server=[your_registry_path] --docker
-username=[username] --docker-password=[token]
```

+

Se você excluir o namespace depois que o segredo é gerado, recrie o namespace e, em seguida, regenere o segredo para o namespace.

   a. Crie o `netapp-acc` namespace (ou nome personalizado).

```
kubectl create ns [netapp-acc or custom namespace]
```

   b. Crie um segredo para o `netapp-acc` namespace (ou nome personalizado). Adicione informações do Docker e execute um dos comandos apropriados, dependendo da sua preferência de Registro:

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-acc or custom namespace] --docker-server=cr.astra.netapp.io --docker-username=[astra_account_id] --docker-password=[astra_api_token]
```

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-acc or custom namespace] --docker-server=[your_registry_path] --docker-username=[username] --docker-password=[token]
```

Instale o operador do Centro de Controle Astra

1. (Apenas registos locais) se estiver a utilizar um registo local, siga estes passos:

   a. Abra a implantação do operador Astra Control Center YAML:

   ```
   vim astra_control_center_operator_deploy.yaml
   ```

   Uma amostra anotada YAML segue estes passos.

   b. Se você usar um Registro que requer autenticação, substitua a linha padrão de `imagePullSecrets: []` pelo seguinte:

   ```
   imagePullSecrets: [{name: astra-registry-cred}]
   ```

   c. Altere `ASTRA_IMAGE_REGISTRY` para a `kube-rbac-proxy` imagem para o caminho do registo onde as imagens foram empurradas para um passo anterior.

   d. Altere `ASTRA_IMAGE_REGISTRY` para a `acc-operator-controller-manager` imagem para o caminho do registo onde as imagens foram empurradas para um passo anterior.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  strategy:
    type: Recreate
```

```
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: ASTRA_IMAGE_REGISTRY/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_INSTALLTIMEOUT
          value: 5m
        image: ASTRA_IMAGE_REGISTRY/acc-operator:24.02.68
        imagePullPolicy: IfNotPresent
        livenessProbe:
          httpGet:
            path: /healthz
            port: 8081
          initialDelaySeconds: 15
          periodSeconds: 20
        name: manager
        readinessProbe:
          httpGet:
            path: /readyz
            port: 8081
          initialDelaySeconds: 5
          periodSeconds: 10
        resources:
          limits:
            cpu: 300m
            memory: 750Mi
          requests:
            cpu: 100m
```

```
            memory: 75Mi
          securityContext:
            allowPrivilegeEscalation: false
        imagePullSecrets: []
        securityContext:
          runAsUser: 65532
        terminationGracePeriodSeconds: 10
```

2. Instale o operador do Centro de Controle Astra:

```
kubectl apply -f astra_control_center_operator_deploy.yaml
```

Expandir para resposta da amostra:

```
namespace/netapp-acc-operator created
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.as
tra.netapp.io created
role.rbac.authorization.k8s.io/acc-operator-leader-election-role
created
clusterrole.rbac.authorization.k8s.io/acc-operator-manager-role
created
clusterrole.rbac.authorization.k8s.io/acc-operator-metrics-reader
created
clusterrole.rbac.authorization.k8s.io/acc-operator-proxy-role
created
rolebinding.rbac.authorization.k8s.io/acc-operator-leader-election-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-manager-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-proxy-
rolebinding created
configmap/acc-operator-manager-config created
service/acc-operator-controller-manager-metrics-service created
deployment.apps/acc-operator-controller-manager created
```

3. Verifique se os pods estão em execução:

```
kubectl get pods -n netapp-acc-operator
```

Configurar o Astra Control Center

1. Edite o arquivo de recursos personalizados (CR) do Astra Control Center

(`astra_control_center.yaml`) para criar contas, suporte, Registro e outras configurações necessárias:

```
vim astra_control_center.yaml
```

Uma amostra anotada YAML segue estes passos.

2. Modifique ou confirme as seguintes definições:

AccountName

Definição	Orientação	Tipo	Exemplo
`accountName`	Altere a `accountName` cadeia de carateres para o nome que deseja associar à conta Astra Control Center. Só pode haver uma accountName.	cadeia de carateres	`Example`

AstraVersion

Definição	Orientação	Tipo	Exemplo
`astraVersion`	A versão do Astra Control Center para implantação. Não é necessária nenhuma ação para esta definição, uma vez que o valor será pré-preenchido.	cadeia de carateres	`24.02.0-69`

Endereço de e-mail

Definição	Orientação	Tipo	Exemplo
`astraAddress`	Altere a `astraAddress` cadeia de carateres para o endereço FQDN (recomendado) ou IP que você deseja usar em seu navegador para acessar o Astra Control Center. Esse endereço define como o Astra Control Center será encontrado em seu data center e será o mesmo FQDN ou endereço IP que você provisionou do balanceador de carga quando concluir "Requisitos do Astra Control Center". NOTA: Não use `http://` nem `https://` no endereço. Copie este FQDN para uso em um passo posterior.	cadeia de carateres	`astra.example.com`

AutoSupport

Suas seleções nesta seção determinam se você participará do aplicativo de suporte Pro-ativo da NetApp, do Consultor Digital e onde os dados são enviados. É necessária uma ligação à Internet (porta 442) e todos os dados de suporte são anonimizados.

Definição	Utilização	Orientação	Tipo	Exemplo
`autoSupport.enrolled`	`enrolled`Os campos ou `url` têm de ser selecionados	Alterar `enrolled` para AutoSupport para `false` sites sem conetividade com a Internet ou manter `true` para sites conetados. Uma configuração de `true` permite que dados anônimos sejam enviados para o NetApp para fins de suporte. A eleição padrão é `false` e indica que nenhum dado de suporte será enviado para o NetApp.	Booleano	`false` (este valor é o padrão)
`autoSupport.url`	`enrolled`Os campos ou `url` têm de ser selecionados	Esta URL determina onde os dados anônimos serão enviados.	cadeia de carateres	`https://support.netapp.com/asupprod/post/1.0/postAsup`

e-mail

Definição	Orientação	Tipo	Exemplo
`email`	Altere a `email` cadeia de carateres para o endereço de administrador inicial padrão. Copie este endereço de e-mail para uso em um passo posterior. Este endereço de e-mail será usado como o nome de usuário da conta inicial para fazer login na IU e será notificado de eventos no Astra Control.	cadeia de carateres	`admin@example.com`

Nome próprio

Definição	Orientação	Tipo	Exemplo
`firstName`	O primeiro nome do administrador inicial padrão associado à conta Astra. O nome usado aqui será visível em um cabeçalho na IU após seu primeiro login.	cadeia de carateres	`SRE`

Sobrenome

Definição	Orientação	Tipo	Exemplo
`lastName`	O sobrenome do administrador inicial padrão associado à conta Astra. O nome usado aqui será visível em um cabeçalho na IU após seu primeiro login.	cadeia de carateres	`Admin`

ImageRegistry

Suas seleções nesta seção definem o Registro de imagem de contentor que hospeda as imagens do aplicativo Astra, o Operador do Centro de Controle Astra e o repositório do Astra Control Center Helm.

Definição	Utilização	Orientação	Tipo	Exemplo
`imageRegistry.name`	Obrigatório	O nome do Registro de imagem Astra Control que hospeda todas as imagens necessárias para implantar o Astra Control Center. O valor será pré-preenchido e nenhuma ação é necessária, a menos que você tenha configurado um Registro local. Para um registo local, substitua este valor existente pelo nome do registo de imagens onde as imagens foram enviadas para o passo anterior. Não utilize `http://` ou `https://` no nome do registo.	cadeia de carateres	`cr.astra.netapp.io` (padrão) `example.registry.com/astra` (exemplo de registro local)

Definição	Utilização	Orientação	Tipo	Exemplo
`imageRegistry.secret`	Opcional	O nome do segredo do Kubernetes usado para autenticar com o Registro de imagens. O valor será pré-preenchido e nenhuma ação é necessária, a menos que você tenha configurado um Registro local e a cadeia de carateres que você inseriu para esse Registro `imageRegistry.name` requer um segredo. IMPORTANTE: Se você estiver usando um Registro local que não requer autorização, você deve excluir essa `secret` linha dentro `imageRegistry` ou a instalação falhará.	cadeia de carateres	`astra-registry-cred`

StorageClass

Definição	Orientação	Tipo	Exemplo
`storageClass`	Altere o `storageClass` valor de `ontap-gold` para outro recurso storageClass conforme exigido pela instalação. Execute o comando `kubectl get sc` para determinar suas classes de armazenamento configuradas existentes. Uma das classes de armazenamento configuradas pelo Astra Control Provisioner deve ser inserida no arquivo manifest (`astra-control-center-<version>.manifest`) e será usada para PVS Astra. Se não estiver definida, a classe de armazenamento padrão será usada. Nota: Se uma classe de armazenamento padrão estiver configurada, certifique-se de que é a única classe de armazenamento que tem a anotação padrão.	cadeia de carateres	`ontap-gold`

VolumeReclaimPolicy

Definição	Orientação	Tipo	Opções
`volumeReclaimPolicy`	Isso define a política de recuperação para PVS do Astra. Definir essa política para `Retain` reter volumes persistentes depois que o Astra for excluído. Definir essa política para `Delete` excluir volumes persistentes depois que o astra for excluído. Se este valor não for definido, os PVS são retidos.	cadeia de carateres	• `Retain` (Este é o valor padrão) • `Delete`

Tipo de ingressType

Definição	Orientação	Tipo	Opções
`ingressType`	Use um dos seguintes tipos de ingresso: **Generic** (`ingressType: "Generic"`) (padrão) Use esta opção quando tiver outro controlador de ingresso em uso ou preferir usar seu próprio controlador de ingresso. Depois que o Astra Control Center for implantado, você precisará configurar o "controlador de entrada" para expor o Astra Control Center com um URL. IMPORTANTE: Se você pretende usar uma malha de serviço com o Astra Control Center, você deve `Generic` selecionar como tipo de ingresso e configurar o seu próprio "controlador de entrada". **AccTraefik** (`ingressType: "AccTraefik"`) Utilize esta opção quando preferir não configurar um controlador de entrada. Isso implanta o gateway Astra Control Center `traefik` como um serviço do tipo Kubernetes LoadBalancer. O Astra Control Center usa um serviço do tipo "LoadBalancer" (`svc/traefik` no namespace Astra Control Center) e exige que seja atribuído um endereço IP externo acessível. Se os balanceadores de carga forem permitidos em seu ambiente e você ainda não tiver um configurado, você poderá usar o MetalLB ou outro balanceador de carga de serviço	cadeia de carateres	• `Generic` (este é o valor padrão) • `AccTraefik`

ScaleSize

Definição	Orientação	Tipo	Opções
`scaleSize`	Por padrão, o Astra usará alta disponibilidade (HA `scaleSize`) do `Medium`, que implanta a maioria dos serviços no HA e implanta várias réplicas para redundância. Com `scaleSize` as `Small`, o Astra reduzirá o número de réplicas para todos os serviços, exceto para serviços essenciais para reduzir o consumo. Dica: `Medium` As implantações consistem em cerca de 100 pods (não incluindo cargas de trabalho transitórias. os pods do 100 são baseados em uma configuração de três nós mestre e três nós de trabalho). Esteja ciente das restrições de limite de rede por pod que podem ser um problema em seu ambiente, especialmente ao considerar cenários de recuperação de desastres.	cadeia de carateres	• `Small` • `Medium` (Este é o valor padrão)

AstraResourcesScaler

Definição	Orientação	Tipo	Opções
`astraResourcesScaler`	Opções de escala para os limites de recursos do AstraControlCenter. Por padrão, o Astra Control Center é implantado com solicitações de recursos definidas para a maioria dos componentes no Astra. Essa configuração permite que a pilha de software Astra Control Center tenha melhor desempenho em ambientes com maior carga e escalabilidade de aplicações. No entanto, em cenários que usam clusters de desenvolvimento ou teste menores, o campo CR `astraResourcesScalar` pode ser definido como `Off`. Isso desativa as solicitações de recursos e permite a implantação em clusters menores.	cadeia de carateres	• `Default` (Este é o valor padrão) • `Off`

Valores adicionais

Adicione os seguintes valores adicionais ao Astra Control Center CR para evitar um problema conhecido na instalação:

```
additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
```

crds

Suas seleções nesta seção determinam como o Astra Control Center deve lidar com CRDs.

Definição	Orientação	Tipo	Exemplo
`crds.externalCertManager`	Se você usar um gerenciador cert externo, `externalCertManager` altere para `true`. O padrão `false` faz com que o Astra Control Center instale seus próprios CRDs de gerenciador de cert durante a instalação. CRDs são objetos de todo o cluster e instalá-los pode ter um impactos em outras partes do cluster. Você pode usar esse sinalizador para sinalizar para o Astra Control Center que essas CRDs serão instaladas e gerenciadas pelo administrador do cluster fora do Astra Control Center.	Booleano	`False` (este valor é o padrão)
`crds.externalTraefik`	Por padrão, o Astra Control Center instalará CRDs Traefik necessários. CRDs são objetos de todo o cluster e instalá-los pode ter um impactos em outras partes do cluster. Você pode usar esse sinalizador para sinalizar para o Astra Control Center que essas CRDs serão instaladas e gerenciadas pelo administrador do cluster fora do Astra Control Center.	Booleano	`False` (este valor é o padrão)

Certifique-se de que selecionou a classe de armazenamento e o tipo de entrada corretos para a sua configuração antes de concluir a instalação.

amostra astra_control_center.yaml

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[cr.astra.netapp.io or your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  scaleSize: "Medium"
  astraResourcesScaler: "Default"
  additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
  crds:
    externalTraefik: false
    externalCertManager: false
```

Instalação completa do operador e do Centro de Controle Astra

1. Se você ainda não fez isso em uma etapa anterior, crie o `netapp-acc` namespace (ou personalizado):

   ```
   kubectl create ns [netapp-acc or custom namespace]
   ```

2. Se você estiver usando uma malha de serviço com o Astra Control Center, adicione a seguinte etiqueta ao `netapp-acc` namespace ou personalizado:

Seu tipo de ingresso (`ingressType`) deve ser definido como `Generic` no Astra Control Center CR antes de prosseguir com este comando.

```
kubectl label ns [netapp-acc or custom namespace] istio-
injection:enabled
```

3. (Recomendado) "Ativar MTLS estritos" para malha de serviço do Istio:

```
kubectl apply -n istio-system -f - <<EOF
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: default
spec:
  mtls:
    mode: STRICT
EOF
```

4. Instale o Astra Control Center no `netapp-acc` namespace (ou personalizado):

```
kubectl apply -f astra_control_center.yaml -n [netapp-acc or custom
namespace]
```

O operador do Astra Control Center executará uma verificação automática dos requisitos de ambiente. A falta "requisitos"pode fazer com que a instalação falhe ou o Astra Control Center não funcione corretamente. próxima seçãoConsulte para verificar se existem mensagens de aviso relacionadas com a verificação automática do sistema.

Verifique o status do sistema

Você pode verificar o status do sistema usando comandos kubectl. Se você preferir usar OpenShift, você pode usar comandos oc comparáveis para etapas de verificação.

Passos

1. Verifique se o processo de instalação não produziu mensagens de avisos relacionadas às verificações de validação:

```
kubectl get acc [astra or custom Astra Control Center CR name] -n
[netapp-acc or custom namespace] -o yaml
```

Mensagens de aviso adicionais também são relatadas nos logs do operador do Centro de Controle Astra.

2. Corrija quaisquer problemas com seu ambiente que foram relatados pelas verificações automatizadas de requisitos.

   Você pode corrigir problemas garantindo que seu ambiente atenda ao do "requisitos" para Astra Control Center.

3. Verifique se todos os componentes do sistema foram instalados com êxito.

   ```
   kubectl get pods -n [netapp-acc or custom namespace]
   ```

   Cada pod deve ter um status de `Running`. Pode levar alguns minutos até que os pods do sistema sejam implantados.

Expandir para resposta de amostra

```
acc-helm-repo-5bd77c9ddd-8wxm2              1/1     Running     0
1h
activity-5bb474dc67-8l9ss                   1/1     Running     0
1h
activity-5bb474dc67-qbrtq                   1/1     Running     0
1h
api-token-authentication-6wbj2              1/1     Running     0
1h
api-token-authentication-9pgw6              1/1     Running     0
1h
api-token-authentication-tqf6d              1/1     Running     0
1h
asup-5495f44dbd-z4kft                       1/1     Running     0
1h
authentication-6fdd899858-5x45s             1/1     Running     0
1h
bucketservice-84d47487d-n9xgp               1/1     Running     0
1h
bucketservice-84d47487d-t5jhm               1/1     Running     0
1h
cert-manager-5dcb7648c4-hbldc               1/1     Running     0
1h
cert-manager-5dcb7648c4-nr9qf               1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-bk2tf    1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-pfnck    1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-ngz2x        1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-rwtbn        1/1     Running     0
1h
certificates-5f5b7b4dd-52tnj                1/1     Running     0
1h
certificates-5f5b7b4dd-gtjbx                1/1     Running     0
1h
certificates-expiry-check-28477260-dz5vw    0/1     Completed   0
1h
cloud-extension-6f58cc579c-lzfmv            1/1     Running     0
1h
cloud-extension-6f58cc579c-zw2km            1/1     Running     0
1h
cluster-orchestrator-79dd5c8d95-qjg92       1/1     Running     0
1h
```

```
composite-compute-85dc84579c-nz82f          1/1     Running     0
1h
composite-compute-85dc84579c-wx2z2          1/1     Running     0
1h
composite-volume-bff6f4f76-789nj            1/1     Running     0
1h
composite-volume-bff6f4f76-kwnd4            1/1     Running     0
1h
credentials-79fd64f788-m7m8f                1/1     Running     0
1h
credentials-79fd64f788-qnc6c                1/1     Running     0
1h
entitlement-f69cdbd77-4p2kn                 1/1     Running     0
1h
entitlement-f69cdbd77-hswm6                 1/1     Running     0
1h
features-7b9585444c-7xd7m                   1/1     Running     0
1h
features-7b9585444c-dcqwc                   1/1     Running     0
1h
fluent-bit-ds-crq8m                         1/1     Running     0
1h
fluent-bit-ds-gmgq8                         1/1     Running     0
1h
fluent-bit-ds-gzr4f                         1/1     Running     0
1h
fluent-bit-ds-j6sf6                         1/1     Running     0
1h
fluent-bit-ds-v4t9f                         1/1     Running     0
1h
fluent-bit-ds-x7j59                         1/1     Running     0
1h
graphql-server-6cc684fb46-2x8lr             1/1     Running     0
1h
graphql-server-6cc684fb46-bshbd             1/1     Running     0
1h
hybridauth-84599f79fd-fjc7k                 1/1     Running     0
1h
hybridauth-84599f79fd-s9pmn                 1/1     Running     0
1h
identity-95df98cb5-dvlmz                    1/1     Running     0
1h
identity-95df98cb5-krf59                    1/1     Running     0
1h
influxdb2-0                                 1/1     Running     0
1h
```

```
keycloak-operator-6d4d688697-cfq8b          1/1     Running     0
1h
krakend-5d5c8f4668-7bq8g                    1/1     Running     0
1h
krakend-5d5c8f4668-t8hbn                    1/1     Running     0
1h
license-689cdd4595-2gsc8                    1/1     Running     0
1h
license-689cdd4595-g6vwk                    1/1     Running     0
1h
login-ui-57bb599956-4fwgz                   1/1     Running     0
1h
login-ui-57bb599956-rhztb                   1/1     Running     0
1h
loki-0                                      1/1     Running     0
1h
metrics-facade-846999bdd4-f7jdm             1/1     Running     0
1h
metrics-facade-846999bdd4-lnsxl             1/1     Running     0
1h
monitoring-operator-6c9d6c4b8c-ggkrl        2/2     Running     0
1h
nats-0                                      1/1     Running     0
1h
nats-1                                      1/1     Running     0
1h
nats-2                                      1/1     Running     0
1h
natssync-server-6df7d6cc68-9v2gd            1/1     Running     0
1h
nautilus-64b7fbdd98-bsgwb                   1/1     Running     0
1h
nautilus-64b7fbdd98-djlhw                   1/1     Running     0
1h
openapi-864584bccc-75nlv                    1/1     Running     0
1h
openapi-864584bccc-zh6bx                    1/1     Running     0
1h
polaris-consul-consul-server-0              1/1     Running     0
1h
polaris-consul-consul-server-1              1/1     Running     0
1h
polaris-consul-consul-server-2              1/1     Running     0
1h
polaris-keycloak-0                          1/1     Running     2 (1h
ago)      1h
```

```
polaris-keycloak-1                          1/1     Running     0
1h
polaris-keycloak-db-0                       1/1     Running     0
1h
polaris-keycloak-db-1                       1/1     Running     0
1h
polaris-keycloak-db-2                       1/1     Running     0
1h
polaris-mongodb-0                           1/1     Running     0
1h
polaris-mongodb-1                           1/1     Running     0
1h
polaris-mongodb-2                           1/1     Running     0
1h
polaris-ui-66476dcf87-f6s8j                 1/1     Running     0
1h
polaris-ui-66476dcf87-ztjk7                 1/1     Running     0
1h
polaris-vault-0                             1/1     Running     0
1h
polaris-vault-1                             1/1     Running     0
1h
polaris-vault-2                             1/1     Running     0
1h
public-metrics-bfc4fc964-x4m79              1/1     Running     0
1h
storage-backend-metrics-7dbb88d4bc-g78cj    1/1     Running     0
1h
storage-provider-5969b5df5-hjvcm            1/1     Running     0
1h
storage-provider-5969b5df5-r79ld            1/1     Running     0
1h
task-service-5fc9dc8d99-4q4f4               1/1     Running     0
1h
task-service-5fc9dc8d99-815zl               1/1     Running     0
1h
task-service-task-purge-28485735-fdzkd      1/1     Running     0
12m
telegraf-ds-2rgm4                           1/1     Running     0
1h
telegraf-ds-4qp6r                           1/1     Running     0
1h
telegraf-ds-77frs                           1/1     Running     0
1h
telegraf-ds-bc725                           1/1     Running     0
1h
```

```
telegraf-ds-cvmxf                          1/1     Running     0
1h
telegraf-ds-tqzgj                          1/1     Running     0
1h
telegraf-rs-5wtd8                          1/1     Running     0
1h
telemetry-service-6747866474-5djnc         1/1     Running     0
1h
telemetry-service-6747866474-thb7r         1/1     Running     1 (1h
ago)      1h
tenancy-5669854fb6-gzdzf                   1/1     Running     0
1h
tenancy-5669854fb6-xvsm2                   1/1     Running     0
1h
traefik-8f55f7d5d-4lgfw                    1/1     Running     0
1h
traefik-8f55f7d5d-j4wt6                    1/1     Running     0
1h
traefik-8f55f7d5d-p6gcq                    1/1     Running     0
1h
trident-svc-7cb5bb4685-54cnq               1/1     Running     0
1h
trident-svc-7cb5bb4685-b28xh               1/1     Running     0
1h
vault-controller-777b9bbf88-b5bqt          1/1     Running     0
1h
vault-controller-777b9bbf88-fdfd8          1/1     Running     0
1h
```

4. (Opcional) Assista os `acc-operator` logs para monitorar o progresso:

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

> ⓘ `accHost` o registro de cluster é uma das últimas operações e, se falhar, não causará falha na implantação. No caso de uma falha de Registro de cluster indicada nos logs, você pode tentar o Registro novamente por meio da "Adicione fluxo de trabalho de cluster na IU" API ou.

5. Quando todos os pods estiverem em execução, verifique se a instalação foi bem-sucedida (`READY` é `True`) e obtenha a senha de configuração inicial que você usará quando fizer login no Astra Control Center:

```
kubectl get AstraControlCenter -n [netapp-acc or custom namespace]
```

Resposta:

```
NAME    UUID                                   VERSION     ADDRESS
READY
astra   9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f  24.02.0-69
10.111.111.111  True
```

Copie o valor UUID. A palavra-passe é `ACC-` seguida pelo valor UUID (`ACC-[UUID]`ou, neste exemplo, `ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f`).

Configure a entrada para o balanceamento de carga

Você pode configurar uma controladora de ingresso do Kubernetes que gerencia o acesso externo a serviços. Esses procedimentos fornecem exemplos de configuração para um controlador de entrada se você usou o padrão do no recurso personalizado do `ingressType: "Generic"` Astra Control Center (`astra_control_center.yaml`). Não é necessário usar este procedimento se você especificou `ingressType: "AccTraefik"` no recurso personalizado do Astra Control Center (`astra_control_center.yaml`).

Depois que o Astra Control Center for implantado, você precisará configurar o controlador Ingress para expor o Astra Control Center com um URL.

As etapas de configuração diferem dependendo do tipo de controlador de entrada que você usa. O Astra Control Center é compatível com muitos tipos de controlador de entrada. Estes procedimentos de configuração fornecem passos de exemplo para alguns tipos comuns de controlador de entrada.

Antes de começar

- O necessário "controlador de entrada" já deve ser implantado.
- O "classe de entrada" correspondente ao controlador de entrada já deve ser criado.

Etapas para a entrada do Istio

1. Configurar a entrada do Istio.

   Este procedimento pressupõe que o Istio é implantado usando o perfil de configuração "padrão".

2. Reúna ou crie o certificado e o arquivo de chave privada desejados para o Ingress Gateway.

   Você pode usar um certificado assinado pela CA ou autoassinado. O nome comum deve ser o endereço Astra (FQDN).

   Exemplo de comando:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out
tls.crt
```

3. Crie um segredo `tls secret name` do tipo `kubernetes.io/tls` para uma chave privada TLS e um certificado, `istio-system namespace` conforme descrito em segredos TLS.

   Exemplo de comando:

```
kubectl create secret tls [tls secret name] --key="tls.key"
--cert="tls.crt" -n istio-system
```

   O nome do segredo deve corresponder ao `spec.tls.secretName` fornecido no `istio-ingress.yaml` arquivo.

4. Implante um recurso de entrada no `netapp-acc` namespace (ou nome personalizado) usando o tipo de recurso v1 para um esquema (`istio-Ingress.yaml` é usado neste exemplo):

```
apiVersion: networking.k8s.io/v1
kind: IngressClass
metadata:
  name: istio
spec:
  controller: istio.io/ingress-controller
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: istio
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: traefik
            port:
              number: 80
```

5. Aplicar as alterações:

```
kubectl apply -f istio-Ingress.yaml
```

6. Verifique o estado da entrada:

```
kubectl get ingress -n [netapp-acc or custom namespace]
```

Resposta:

```
NAME    CLASS HOSTS             ADDRESS         PORTS   AGE
ingress istio astra.example.com 172.16.103.248  80, 443 1h
```

7. Concluir a instalação do Astra Control Center.

Etapas para o controlador nginx Ingress

1. Crie um segredo do tipo `kubernetes.io/tls` para uma chave privada TLS e um certificado no `netapp-acc` namespace (ou nome personalizado), conforme descrito em "Segredos TLS".
2. Implantar um recurso de entrada no `netapp-acc` namespace (ou nome personalizado) usando o tipo de recurso v1 para um esquema (`nginx-Ingress.yaml` é usado neste exemplo):

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netapp-acc-ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: [class name for nginx controller]
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: <ACC address>
    http:
      paths:
        - path:
          backend:
            service:
              name: traefik
              port:
                number: 80
          pathType: ImplementationSpecific
```

3. Aplicar as alterações:

```
kubectl apply -f nginx-Ingress.yaml
```

O NetApp recomenda a instalação do controlador nginx como uma implementação em vez de um `daemonSet`.

Passos para o controlador OpenShift Ingress

1. Procure seu certificado e prepare os arquivos de chave, certificado e CA para uso pela rota OpenShift.
2. Crie a rota OpenShift:

```
oc create route edge --service=traefik --port=web -n [netapp-acc or
custom namespace] --insecure-policy=Redirect --hostname=<ACC address>
--cert=cert.pem --key=key.pem
```

Faça login na IU do Astra Control Center

Depois de instalar o Astra Control Center, você alterará a senha do administrador padrão e fará login no painel da IU do Astra Control Center.

Passos

1. Em um navegador, insira o FQDN (incluindo o `https://` prefixo) usado no no `astraAddress` `astra_control_center.yaml` CR quando Você instalou o Astra Control Center.
2. Aceite os certificados autoassinados, se solicitado.

   Você pode criar um certificado personalizado após o login.

3. Na página de login do Astra Control Center, insira o valor usado `email` no `astra_control_center.yaml` CR quando Você instalou o Astra Control Center, seguido da senha de configuração inicial (`ACC-[UUID]`).

   Se você digitar uma senha incorreta três vezes, a conta de administrador será bloqueada por 15 minutos.

4. Selecione **Login**.
5. Altere a senha quando solicitado.

   Se este for o seu primeiro login e você esquecer a senha e nenhuma outra conta de usuário administrativo ainda tiver sido criada, entre em Contato "Suporte à NetApp" para obter assistência de recuperação de senha.

6. (Opcional) Remova o certificado TLS autoassinado existente e substitua-o por um "Certificado TLS personalizado assinado por uma autoridade de certificação (CA)".

Solucionar problemas da instalação

Se algum dos serviços estiver `Error` no estado, pode inspecionar os registos. Procure códigos de resposta da API na faixa 400 a 500. Eles indicam o lugar onde uma falha aconteceu.

Opções

- Para inspecionar os logs do operador do Centro de Controle Astra, digite o seguinte:

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

- Para verificar a saída do Astra Control Center CR:

```
kubectl get acc -n [netapp-acc or custom namespace] -o yaml
```

Procedimentos alternativos de instalação

- **Instalar com o Red Hat OpenShift OperatorHub**: Use isso "procedimento alternativo" para instalar o Astra Control Center no OpenShift usando o OperatorHub.
- **Instalar na nuvem pública com o Cloud Volumes ONTAP backend**: Use "estes procedimentos" para instalar o Astra Control Center no Amazon Web Services (AWS), no Google Cloud Platform (GCP) ou no Microsoft Azure com um back-end de storage do Cloud Volumes ONTAP.

O que vem a seguir

- (Opcional) dependendo do seu ambiente, conclua a pós-instalação"etapas de configuração".
- "Depois de instalar o Astra Control Center, fazer login na IU e alterar sua senha, você deseja configurar uma licença, adicionar clusters, habilitar a autenticação, gerenciar armazenamento e adicionar buckets".

Configurar um gerenciador de cert externo

Se um gerenciador de cert já existir no cluster do Kubernetes, você precisará executar algumas etapas de pré-requisito para que o Astra Control Center não instale seu próprio gerenciador de cert.

Passos

1. Confirme se você tem um gerenciador cert instalado:

   ```
   kubectl get pods -A | grep 'cert-manager'
   ```

   Resposta da amostra:

   ```
   cert-manager   essential-cert-manager-84446f49d5-sf2zd              1/1
   Running    0      6d5h
   cert-manager   essential-cert-manager-cainjector-66dc99cc56-9ldmt   1/1
   Running    0      6d5h
   cert-manager   essential-cert-manager-webhook-56b76db9cc-fjqrq      1/1
   Running    0      6d5h
   ```

2. Crie um par de certificados/chaves para o `astraAddress` FQDN:

   ```
   openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out
   tls.crt
   ```

   Resposta da amostra:

```
Generating a 2048 bit RSA private key
..................+++
.......................................+++
writing new private key to 'tls.key'
```

3. Crie um segredo com arquivos gerados anteriormente:

```
kubectl create secret tls selfsigned-tls --key tls.key --cert tls.crt -n
<cert-manager-namespace>
```

Resposta da amostra:

```
secret/selfsigned-tls created
```

4. Crie um `ClusterIssuer` arquivo que seja **exatamente** a seguir, mas inclua o local do namespace onde seus `cert-manager` pods estão instalados:

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: astra-ca-clusterissuer
  namespace: <cert-manager-namespace>
spec:
  ca:
    secretName: selfsigned-tls
```

```
kubectl apply -f ClusterIssuer.yaml
```

Resposta da amostra:

```
clusterissuer.cert-manager.io/astra-ca-clusterissuer created
```

5. Verifique se o `ClusterIssuer` foi apresentado corretamente. `Ready` deve ser `True` antes que você possa prosseguir:

```
kubectl get ClusterIssuer
```

Resposta da amostra:

```
NAME                        READY   AGE
astra-ca-clusterissuer      True    9s
```

6. Preencha "Processo de instalação do Astra Control Center"o . Há um "Etapa de configuração necessária para o cluster Astra Control Center YAML" em que você altera o valor CRD para indicar que o gerenciador cert está instalado externamente. Você deve concluir esta etapa durante a instalação para que o Astra Control Center reconheça o gerenciador de cert externo.

Instale o Astra Control Center usando o OpenShift OperatorHub

Se você usar o Red Hat OpenShift, poderá instalar o Astra Control Center usando o operador certificado Red Hat. Use este procedimento para instalar o Astra Control Center a partir do "Catálogo de ecossistemas da Red Hat" ou usando o Red Hat OpenShift Container Platform.

Depois de concluir este procedimento, terá de voltar ao procedimento de instalação para concluir o para verificar o "passos restantes"êxito da instalação e iniciar sessão.

Antes de começar

- * Atender pré-requisitos ambientais *"Antes de começar a instalação, prepare seu ambiente para a implantação do Astra Control Center": .

  Implante o Astra Control Center em um domínio de terceiros ou local secundário. Isso é recomendado para replicação de aplicativos e recuperação de desastres aprimorada.

- **Garanta operadores de cluster e serviços de API saudáveis**:
  - A partir do cluster OpenShift, certifique-se de que todos os operadores de cluster estão em um estado saudável:

    ```
    oc get clusteroperators
    ```

  - A partir do cluster OpenShift, certifique-se de que todos os serviços de API estão em um estado saudável:

    ```
    oc get apiservices
    ```

- **Certifique-se de que um FQDN roteável**: O FQDN Astra que você planeja usar pode ser roteado para o cluster. Isso significa que você tem uma entrada DNS no seu servidor DNS interno ou está usando uma rota URL principal que já está registrada.
- * Obter permissões OpenShift*: Você precisará de todas as permissões necessárias e acesso à Red Hat OpenShift Container Platform para executar as etapas de instalação descritas.
- **Configurar um gerenciador cert**: Se um gerenciador cert já existir no cluster, você precisará executar alguns "etapas de pré-requisito" para que o Astra Control Center não instale seu próprio gerenciador cert. Por padrão, o Astra Control Center instala seu próprio gerenciador de cert durante a instalação.
- * Configurar o controlador de entrada do Kubernetes*: Se você tiver uma controladora de entrada do

Kubernetes que gerencia o acesso externo a serviços, como balanceamento de carga em um cluster, será necessário configurá-la para uso com o Astra Control Center:

a. Crie o namespace do operador:

```
oc create namespace netapp-acc-operator
```

b. "Conclua a configuração" para o seu tipo de controlador de entrada.

- **(somente driver SAN ONTAP) Ativar multipath**: Se você estiver usando um driver SAN ONTAP, verifique se o multipath está habilitado em todos os clusters Kubernetes.

Você também deve considerar o seguinte:

- **Tenha acesso ao Registro de imagens do NetApp Astra Control**:

  Você tem a opção de obter imagens de instalação e melhorias de funcionalidade para o Astra Control, como o Astra Control Provisioner, a partir do Registro de imagens do NetApp.

  a. Registre seu ID de conta Astra Control que você precisará fazer login no Registro.

     Você pode ver o ID da conta na IU da Web do Astra Control Service. Selecione o ícone de figura no canto superior direito da página, selecione **Acesso à API** e anote o ID da sua conta.

  b. 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.

  c. Faça login no Registro do Astra Control:

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

- **Instalar uma malha de serviço para comunicações seguras**: É altamente recomendável que os canais de comunicação do cluster de host Astra Control sejam protegidos usando um "malha de serviço suportada".

  > ⓘ A integração do Astra Control Center com uma malha de serviço só pode ser feita durante o Astra Control Center "instalação"e não independente desse processo. A alteração de um ambiente de malha para um ambiente sem malha não é suportada.

  Para uso em malha de serviço do Istio, você precisará fazer o seguinte:

  - Adicione `istio-injection:enabled` um rótulo ao namespace Astra antes de implantar o Astra Control Center.
  - Utilize o `Generic` definição de entrada e forneça uma entrada alternativa para "balanceamento de carga externo"o .
  - Para clusters do Red Hat OpenShift, você precisará definir `NetworkAttachmentDefinition` em todos os namespaces associados do Astra Control Center , `netapp-monitoring` para clusters de aplicativos ou quaisquer(`netapp-acc-operator` namespaces `netapp-acc` personalizados que tenham sido substituídos).

```
cat <<EOF | oc -n netapp-acc-operator create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-acc create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-monitoring create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF
```

Passos

- Faça download e extraia Astra Control Center
- Conclua as etapas adicionais se você usar um Registro local
- Localize a página de instalação do operador
- Instale o operador
- Instale o Astra Control Center

> Não exclua o operador Astra Control Center (por exemplo, `kubectl delete -f astra_control_center_operator_deploy.yaml`) a qualquer momento durante a instalação ou operação do Astra Control Center para evitar a exclusão de pods.

Faça download e extraia Astra Control Center

Faça o download das imagens do Astra Control Center de um dos seguintes locais:

- **Registro de imagem do Serviço de Controle Astra**: Use esta opção se você não usar um Registro local com as imagens do Centro de Controle Astra ou se preferir esse método para o download do pacote no site de suporte da NetApp.
- **Site de suporte da NetApp**: Use essa opção se você usar um Registro local com as imagens do Centro de Controle Astra.

Registro de imagem Astra Control

1. Faça login no Astra Control Service.
2. No Dashboard, selecione **Deploy a self-managed instance of Astra Control**.
3. Siga as instruções para fazer login no Registro de imagens do Astra Control, extrair a imagem de instalação do Astra Control Center e extrair a imagem.

Site de suporte da NetApp

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

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

```
openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub
-signature certs/astra-control-center-[version].tar.gz.sig astra-
control-center-[version].tar.gz
```

   A saída será `Verified OK` exibida após a verificação bem-sucedida.

3. Extraia as imagens do pacote Astra Control Center:

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

Conclua as etapas adicionais se você usar um Registro local

Se você está planejando enviar o pacote Astra Control Center para o seu Registro local, você precisa usar o plugin de linha de comando NetApp Astra kubectl.

Instale o plug-in NetApp Astra kubectl

Conclua estas etapas para instalar o plugin de linha de comando mais recente do NetApp Astra kubectl.

Antes de começar

O NetApp fornece binários de plug-in para diferentes arquiteturas de CPU e sistemas operacionais. Você precisa saber qual CPU e sistema operacional você tem antes de executar esta tarefa.

Se você já tiver o plugin instalado a partir de uma instalação anterior, "certifique-se de que tem a versão mais recente" antes de concluir estas etapas.

Passos

1. Liste os binários disponíveis do plug-in NetApp Astra kubectl e observe o nome do arquivo que você precisa para o seu sistema operacional e arquitetura de CPU:

A biblioteca de plugins kubectl faz parte do pacote tar e é extraída para a pasta `kubectl-astra`.

```
ls kubectl-astra/
```

2. Mova o binário correto para o caminho atual e renomeie-o para `kubectl-astra`:

```
cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
```

Adicione as imagens ao seu registo

1. Se você estiver planejando enviar o pacote Astra Control Center para o Registro local, conclua a sequência de etapas apropriada para o mecanismo de contêiner:

Docker

a. Mude para o diretório raiz do tarball. Você deve ver o `acc.manifest.bundle.yaml` arquivo e estes diretórios:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Envie as imagens do pacote no diretório de imagens do Astra Control Center para o Registro local. Faça as seguintes substituições antes de executar o `push-images` comando:

- Substitua o <BUNDLE_FILE> pelo nome do arquivo do pacote Astra Control (`acc.manifest.bundle.yaml`).
- Substitua o <MY_FULL_REGISTRY_PATH> pela URL do repositório Docker; por exemplo"<a href="https://<docker-registry>"" class="bare">https://<docker-registry>", .
- Substitua o <MY_REGISTRY_USER> pelo nome de usuário.
- Substitua o <MY_REGISTRY_TOKEN> por um token autorizado para o Registro.

```
kubectl astra packages push-images -m <BUNDLE_FILE> -r
<MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
<MY_REGISTRY_TOKEN>
```

Podman

a. Mude para o diretório raiz do tarball. Você deve ver este arquivo e diretório:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Inicie sessão no seu registo:

```
podman login <YOUR_REGISTRY>
```

c. Prepare e execute um dos seguintes scripts personalizados para a versão do Podman que você usa. Substitua o <MY_FULL_REGISTRY_PATH> pela URL do seu repositório que inclui quaisquer subdiretórios.

```
<strong>Podman 4</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

O caminho da imagem que o script cria deve ser semelhante ao seguinte, dependendo da configuração do Registro:

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/24.02.0-69/image:version
```

2. Altere o diretório:

```
cd manifests
```

Localize a página de instalação do operador

1. Execute um dos seguintes procedimentos para acessar a página de instalação do operador:

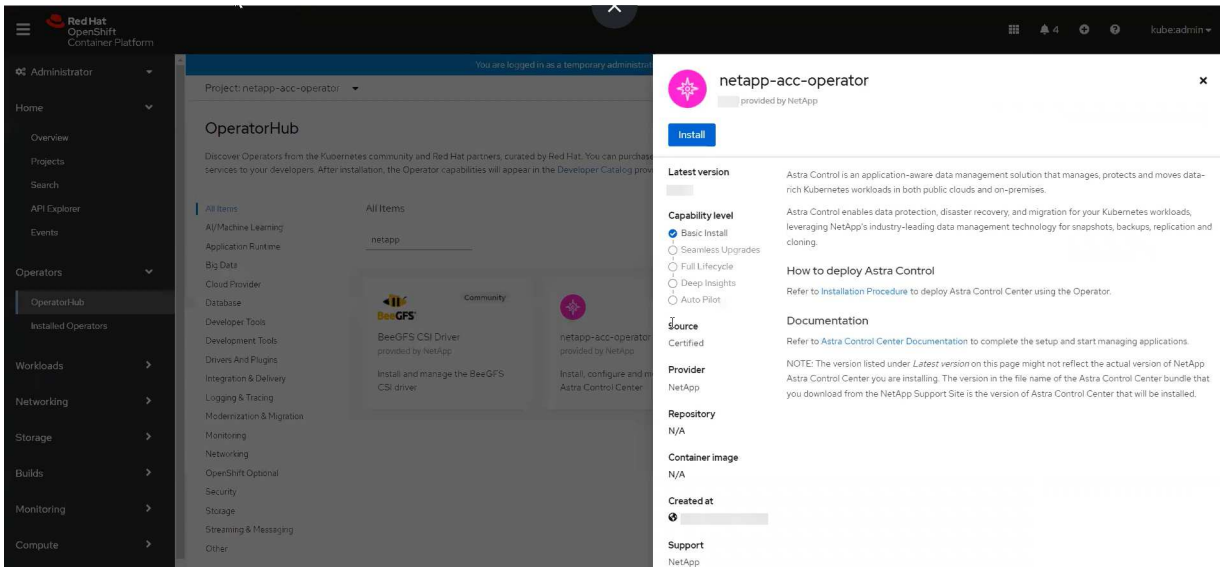
Red Hat OpenShift web console

a. Faça login na IU da OpenShift Container Platform.

b. No menu lateral, selecione **operadores > OperatorHub**.

> Você só pode fazer upgrade para a versão atual do Astra Control Center usando esse operador.

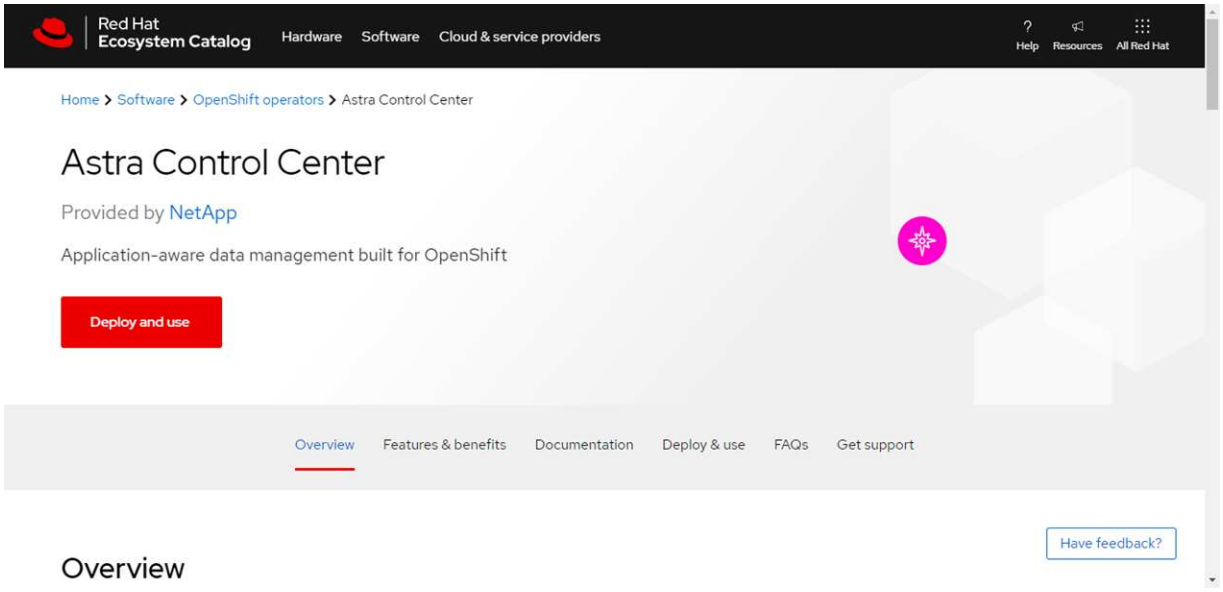
c. Procure `netapp-acc` e selecione o operador do Centro de Controle NetApp Astra.



Catálogo de ecossistemas da Red Hat

a. Selecione o Centro de Controle NetApp Astra "operador" .

b. Selecione **Deploy and use**.



Instale o operador

1. Preencha a página **Instalar Operador** e instale o operador:

   O operador estará disponível em todos os namespaces de cluster.

   a. Selecione o namespace do operador ou `netapp-acc-operator` o namespace será criado automaticamente como parte da instalação do operador.

   b. Selecione uma estratégia de aprovação manual ou automática.

      Recomenda-se a aprovação manual. Você deve ter apenas uma única instância de operador em execução por cluster.

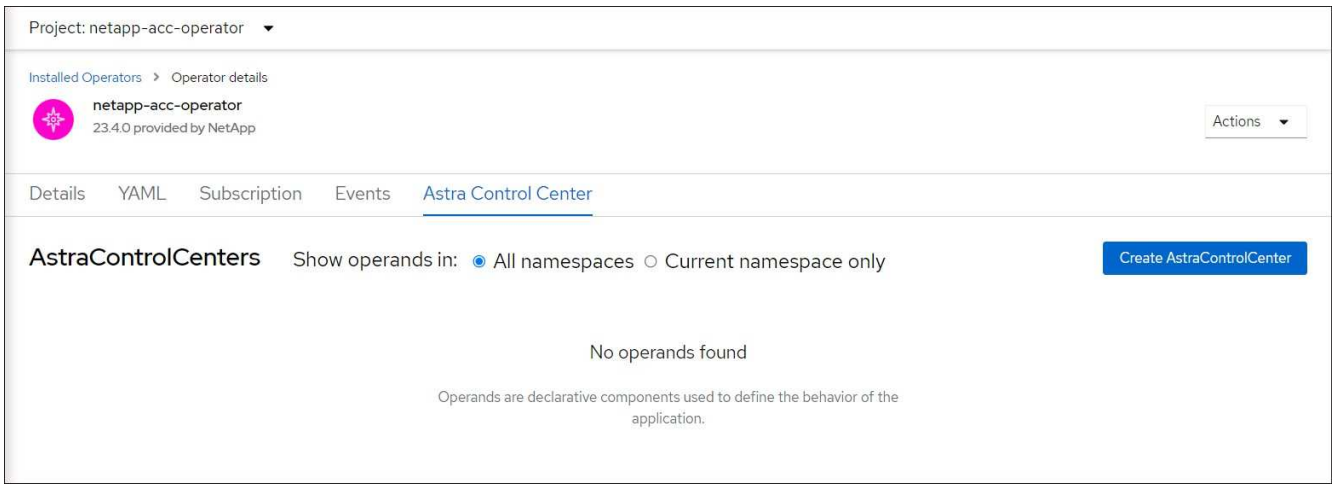
   c. Selecione **Instalar**.

      Se selecionou uma estratégia de aprovação manual, ser-lhe-á pedido que aprove o plano de instalação manual para este operador.

2. No console, vá para o menu OperatorHub e confirme se o operador instalou com êxito.

Instale o Astra Control Center

1. No console dentro da guia **Astra Control Center** do operador Astra Control Center, selecione **Create AstraControlCenter**.



2. Preencha o `Create AstraControlCenter` campo do formulário:

   a. Mantenha ou ajuste o nome do Astra Control Center.

   b. Adicione etiquetas para o Astra Control Center.

   c. Ative ou desative o suporte automático. Recomenda-se a manutenção da funcionalidade de suporte automático.

   d. Insira o FQDN ou o endereço IP do Centro de Controle Astra. Não introduza `http://` ou `https://` no campo de endereço.

   e. Digite a versão do Astra Control Center; por exemplo, 24.02.0-69.

   f. Insira um nome de conta, endereço de e-mail e sobrenome do administrador.

   g. Escolha uma política de recuperação de volume de `Retain` , `Recycle` ou `Delete`. O valor padrão é

`Retain`.

h. Selecione o tamanho da escala da instalação.

> Por padrão, o Astra usará alta disponibilidade (HA `scaleSize`) do `Medium`, que implanta a maioria dos serviços no HA e implanta várias réplicas para redundância. Com `scaleSize` as `Small`, o Astra reduzirá o número de réplicas para todos os serviços, exceto para serviços essenciais para reduzir o consumo.

i. Selecione o tipo de entrada:

- **Generic(`ingressType: "Generic"`)** (predefinição)

  Utilize esta opção quando tiver outro controlador de entrada em utilização ou preferir utilizar o seu próprio controlador de entrada. Depois que o Astra Control Center for implantado, você precisará configurar o "controlador de entrada" para expor o Astra Control Center com um URL.

- **AccTraefik** (`ingressType: "AccTraefik"`)

  Utilize esta opção quando preferir não configurar um controlador de entrada. Isso implanta o gateway Astra Control Center `traefik` como um serviço do tipo "LoadBalancer" do Kubernetes.

O Astra Control Center usa um serviço do tipo "LoadBalancer" (`svc/traefik` no namespace Astra Control Center) e exige que seja atribuído um endereço IP externo acessível. Se os balanceadores de carga forem permitidos em seu ambiente e você ainda não tiver um configurado, você poderá usar o MetalLB ou outro balanceador de carga de serviço externo para atribuir um endereço IP externo ao serviço. Na configuração do servidor DNS interno, você deve apontar o nome DNS escolhido para o Astra Control Center para o endereço IP com balanceamento de carga.

> Para obter detalhes sobre o tipo de serviço "LoadBalancer" e Ingress, "Requisitos"consulte .

a. Em **Image Registry**, use o valor padrão a menos que você tenha configurado um Registro local. Para um registo local, substitua este valor pelo caminho do registo de imagens local onde empurrou as imagens numa etapa anterior. Não introduza `http://` ou `https://` no campo de endereço.

b. Se utilizar um registo de imagens que necessite de autenticação, introduza o segredo da imagem.

> Se você usar um Registro que requer autenticação, crie um segredo no cluster.

c. Introduza o nome do administrador.

d. Configurar o dimensionamento de recursos.

e. Forneça a classe de armazenamento padrão.

> Se uma classe de armazenamento padrão estiver configurada, certifique-se de que é a única classe de armazenamento que tem a anotação padrão.

f. Definir preferências de tratamento de CRD.

3. Selecione a vista YAML para rever as definições selecionadas.
4. `Create`Selecione .

Crie um segredo de Registro

Se você usar um Registro que requer autenticação, crie um segredo no cluster OpenShift e insira o nome secreto no `Create AstraControlCenter` campo formulário.

1. Crie um namespace para o operador Astra Control Center:

```
oc create ns [netapp-acc-operator or custom namespace]
```

2. Crie um segredo neste namespace:

```
oc create secret docker-registry astra-registry-cred -n [netapp-acc-
operator or custom namespace] --docker-server=[your_registry_path]
--docker username=[username] --docker-password=[token]
```

> O Astra Control suporta apenas segredos de registo do Docker.

3. Preencha os campos restantes em O campo criar formulário AstraControlCenter.

O que vem a seguir

Preencha o "passos restantes" para verificar se o Astra Control Center foi instalado com sucesso, configure um controlador de entrada (opcional) e faça login na IU. Além disso, você precisará executar "tarefas de configuração" depois de concluir a instalação.

Instalar o Astra Control Center com um back-end de storage do Cloud Volumes ONTAP

Com o Astra Control Center, você pode gerenciar suas aplicações em um ambiente de nuvem híbrida com clusters Kubernetes autogerenciados e instâncias do Cloud Volumes ONTAP. É possível implantar o Astra Control Center nos clusters de Kubernetes no local ou em um dos clusters de Kubernetes autogerenciado no ambiente de nuvem.

Em uma dessas implantações, você pode executar operações de gerenciamento de dados de aplicações usando o Cloud Volumes ONTAP como um back-end de storage. Você também pode configurar um bucket do S3 como o destino de backup.

Para instalar o Astra Control Center no Amazon Web Services (AWS), no Google Cloud Platform (GCP) e no Microsoft Azure com um back-end de storage do Cloud Volumes ONTAP, execute as etapas a seguir, dependendo do ambiente de nuvem.

- Implante o Astra Control Center na Amazon Web Services
- Implante o Astra Control Center no Google Cloud Platform
- Implante o Astra Control Center no Microsoft Azure

Você pode gerenciar seus aplicativos em distribuições com clusters do Kubernetes autogerenciados, como o OpenShift Container Platform (OCP). Somente clusters de OCP autogeridos são validados para implantar o Astra Control Center.

Implante o Astra Control Center na Amazon Web Services

É possível implantar o Astra Control Center em um cluster Kubernetes autogerenciado hospedado em uma nuvem pública da Amazon Web Services (AWS).

O que você precisará para a AWS

Antes de implantar o Astra Control Center na AWS, você precisará dos seguintes itens:

- Licença do Astra Control Center. "Requisitos de licenciamento do Astra Control Center"Consulte a .
- "Atender aos requisitos do Astra Control Center".
- Conta do NetApp Cloud Central
- Se estiver usando OCP, permissões do Red Hat OpenShift Container Platform (OCP) (no nível do namespace para criar pods)
- Credenciais da AWS, ID de acesso e chave secreta com permissões que permitem criar buckets e conetores
- Acesso e login do AWS Account Elastic Container Registry (ECR)
- A zona hospedada da AWS e a entrada do Amazon Route 53 são necessárias para acessar a IU do Astra Control

Requisitos de ambiente operacional para a AWS

O Astra Control Center requer o seguinte ambiente operacional para a AWS:

- Red Hat OpenShift Container Platform 4,11 a 4,13

Certifique-se de que o ambiente operacional escolhido para hospedar o Astra Control Center atenda aos requisitos básicos de recursos descritos na documentação oficial do ambiente.

O Astra Control Center requer recursos específicos, além dos requisitos de recursos do ambiente. "Requisitos do ambiente operacional do Astra Control Center"Consulte a .

> O token de Registro da AWS expira em 12 horas, após o qual você terá que renovar o segredo de Registro de imagem do Docker.

Visão geral da implantação para AWS

Aqui está uma visão geral do processo de instalação do Astra Control Center for AWS com o Cloud Volumes ONTAP como um back-end de storage.

Cada uma destas etapas é explicada em mais detalhes abaixo.

1. Certifique-se de que tem permissões IAM suficientes.
2. Instale um cluster RedHat OpenShift na AWS.
3. Configurar a AWS.
4. Configure o NetApp BlueXP para AWS.
5. Instalar o Astra Control Center for AWS.

Certifique-se de que tem permissões IAM suficientes

Certifique-se de que você tenha funções e permissões suficientes do IAM que permitam instalar um cluster do RedHat OpenShift e um conetor do NetApp BlueXP (antigo Gerenciador de nuvem).

```
https://docs.netapp.com/us-en/cloud-manager-setup-admin/concept-accounts-
aws.html#initial-aws-credentials["Credenciais iniciais da AWS"^]Consulte .
```

Instale um cluster RedHat OpenShift na AWS

Instale um cluster do RedHat OpenShift Container Platform na AWS.

Para obter instruções de instalação, "Instalar um cluster na AWS no OpenShift Container Platform" consulte .

Configurar a AWS

Em seguida, configure a AWS para criar uma rede virtual, configurar instâncias de computação EC2 e criar um bucket do AWS S3. Se você não puder acessar o Registro de imagem do Centro de Controle Astra do NetApp, também precisará criar um Registro de contêiner elástico (ECR) para hospedar as imagens do Centro de Controle Astra e enviar as imagens para esse Registro.

Siga a documentação da AWS para concluir as etapas a seguir. "Documentação de instalação da AWS"Consulte .

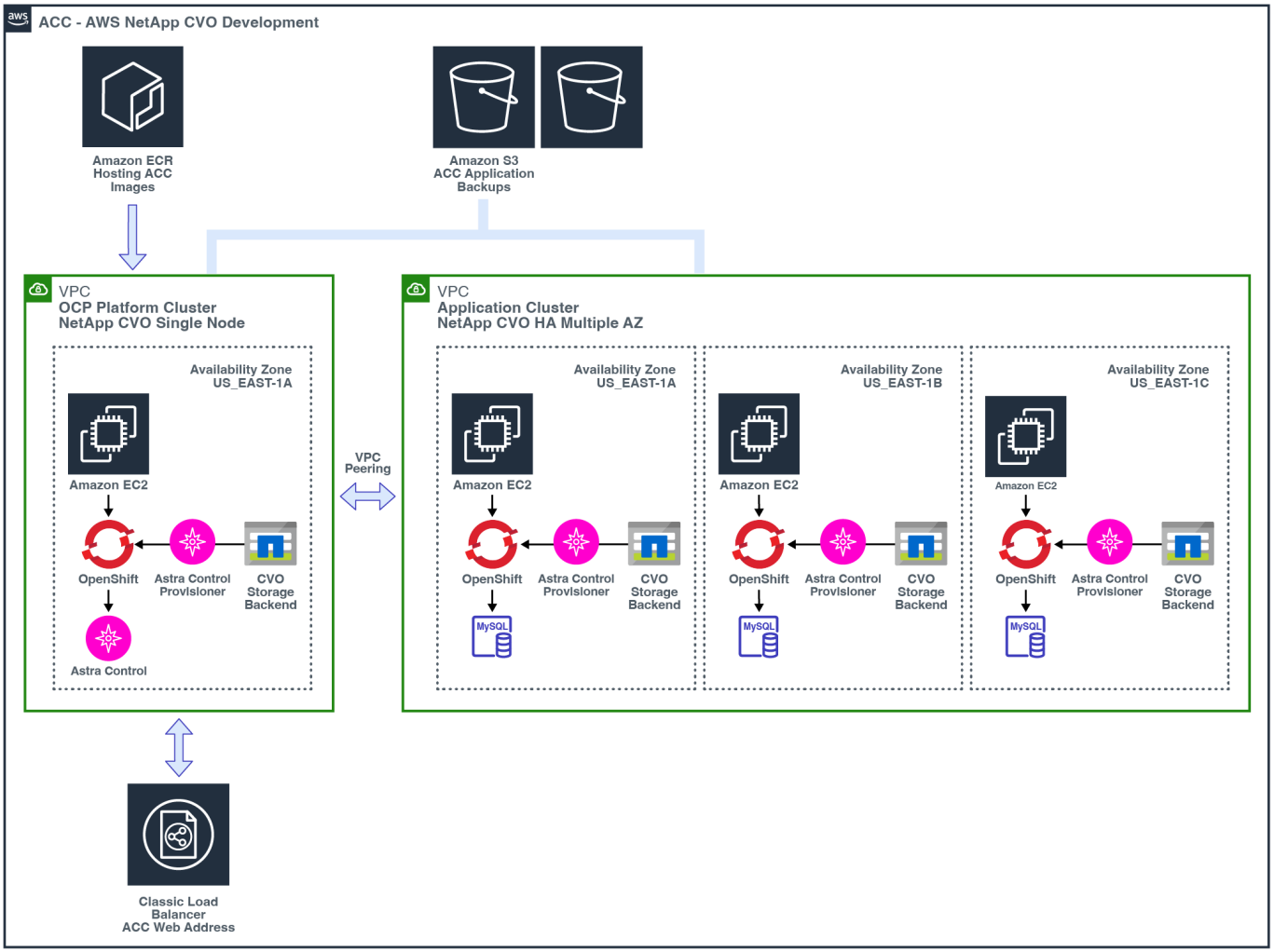
1. Crie uma rede virtual da AWS.
2. Analise as instâncias de computação do EC2. Isso pode ser um servidor bare metal ou VMs na AWS.
3. Se o tipo de instância ainda não corresponder aos requisitos mínimos de recursos do Astra para nós mestres e trabalhadores, altere o tipo de instância na AWS para atender aos requisitos do Astra. "Requisitos do Astra Control Center"Consulte a .
4. Crie pelo menos um bucket do AWS S3 para armazenar seus backups.
5. (Opcional) se você não puder acessar o Registro de imagens do NetApp, faça o seguinte:
   a. Crie um AWS Elastic Container Registry (ECR) para hospedar todas as imagens do Astra Control Center.

      Se você não criar o ECR, o Astra Control Center não poderá acessar dados de monitoramento de um cluster que contém o Cloud Volumes ONTAP com um back-end da AWS. O problema é causado quando o cluster que você tenta descobrir e gerenciar usando o Astra Control Center não tem acesso ao AWS ECR.

   b. Envie as imagens do Astra Control Center para o Registro definido.

O token AWS Elastic Container Registry (ECR) expira após 12 horas e faz com que as operações de clone entre clusters falhem. Esse problema ocorre ao gerenciar um back-end de storage do Cloud Volumes ONTAP configurado para AWS. Para corrigir esse problema, autentique novamente com o ECR e gere um novo segredo para que as operações de clone sejam retomadas com sucesso.

Veja um exemplo de implantação da AWS:



Configure o NetApp BlueXP para AWS

Usando o NetApp BlueXP , crie uma área de trabalho, adicione um conetor à AWS, crie um ambiente de trabalho e importe o cluster.

Siga a documentação do BlueXP para concluir as etapas a seguir. Veja o seguinte:

- "Introdução ao Cloud Volumes ONTAP na AWS".
- "Crie um conetor na AWS usando o BlueXP "

Passos

1. Adicione suas credenciais ao BlueXP .
2. Criar um espaço de trabalho.
3. Adicione um conetor para a AWS. Escolha a AWS como o provedor.
4. Crie um ambiente de trabalho para seu ambiente de nuvem.
    a. Localização: "Amazon Web Services (AWS)"
    b. Tipo: "Cloud Volumes ONTAP HA"
5. Importe o cluster OpenShift. O cluster se conetará ao ambiente de trabalho que você acabou de criar.
    a. Veja os detalhes do cluster do NetApp selecionando **K8s** > **Lista de clusters** > **Detalhes do cluster**.

b. No canto superior direito, observe a versão Astra Control Provisioner.

c. Observe as classes de storage de cluster do Cloud Volumes ONTAP que mostram o NetApp como o provisionador.

   Isso importa seu cluster Red Hat OpenShift e atribui a ele uma classe de armazenamento padrão. Você seleciona a classe de armazenamento. O Astra Control Provisioner é instalado automaticamente como parte do processo de importação e descoberta.

6. Observe todos os volumes e volumes persistentes nessa implantação do Cloud Volumes ONTAP.

O Cloud Volumes ONTAP pode operar como um único nó ou em alta disponibilidade. Se a HA estiver ativada, observe o status da HA e o status da implantação do nó em execução na AWS.

Instalar o Astra Control Center for AWS

Siga o padrão "Instruções de instalação do Astra Control Center".

A AWS usa o tipo de bucket Generic S3.

Implante o Astra Control Center no Google Cloud Platform

É possível implantar o Astra Control Center em um cluster autogerenciado do Kubernetes hospedado em uma nuvem pública do Google Cloud Platform (GCP).

O que você precisará para o GCP

Antes de implantar o Astra Control Center no GCP, você precisará dos seguintes itens:

- Licença do Astra Control Center. "Requisitos de licenciamento do Astra Control Center"Consulte a .
- "Atender aos requisitos do Astra Control Center".
- Conta do NetApp Cloud Central
- Se estiver usando OCP, Red Hat OpenShift Container Platform (OCP) 4,11 a 4,13
- Se estiver usando OCP, permissões do Red Hat OpenShift Container Platform (OCP) (no nível do namespace para criar pods)
- Conta de serviço do GCP com permissões que permitem criar buckets e conetores

Requisitos do ambiente operacional do GCP

Certifique-se de que o ambiente operacional escolhido para hospedar o Astra Control Center atenda aos requisitos básicos de recursos descritos na documentação oficial do ambiente.

O Astra Control Center requer recursos específicos, além dos requisitos de recursos do ambiente. "Requisitos do ambiente operacional do Astra Control Center"Consulte a .

Visão geral da implantação do GCP

Veja a seguir uma visão geral do processo de instalação do Astra Control Center em um cluster de OCP autogerenciado no GCP, com o Cloud Volumes ONTAP como um back-end de storage.

Cada uma destas etapas é explicada em mais detalhes abaixo.

1. Instale um cluster RedHat OpenShift no GCP.
2. Crie um projeto do GCP e uma nuvem privada virtual.
3. Certifique-se de que tem permissões IAM suficientes.
4. Configurar o GCP.
5. Configurar o NetApp BlueXP para GCP.
6. Instalar o Astra Control Center no GCP.

Instale um cluster RedHat OpenShift no GCP

A primeira etapa é instalar um cluster do RedHat OpenShift no GCP.

Para obter instruções de instalação, consulte o seguinte:

- "Instalação de um cluster OpenShift no GCP"
- "Criando uma conta de serviço do GCP"

Crie um projeto do GCP e uma nuvem privada virtual

Crie pelo menos um projeto do GCP e a Virtual Private Cloud (VPC).

OpenShift pode criar seus próprios grupos de recursos. Além disso, você também deve definir uma VPC do GCP. Consulte a documentação do OpenShift.

Você pode querer criar um grupo de recursos de cluster de plataforma e um grupo de recursos de cluster OpenShift de aplicativo de destino.

Certifique-se de que tem permissões IAM suficientes

Certifique-se de que você tenha funções e permissões suficientes do IAM que permitam instalar um cluster do RedHat OpenShift e um conetor do NetApp BlueXP (antigo Gerenciador de nuvem).

```
https://docs.netapp.com/us-en/cloud-manager-setup-admin/task-creating-
connectors-gcp.html#setting-up-permissions["Credenciais e permissões
iniciais do GCP"^]Consulte .
```

Configurar o GCP

Em seguida, configure o GCP para criar uma VPC, configurar instâncias de computação e criar um Google Cloud Object Storage. Se você não puder acessar o Registro de imagens do NetApp, também precisará criar um Registro de contentor do Google para hospedar as imagens do Centro de Controle Astra e enviar as imagens para esse Registro.

Siga a documentação do GCP para concluir as etapas a seguir. Consulte Instalando o cluster OpenShift no GCP.

1. Crie um projeto do GCP e uma VPC no GCP que você planeja usar para o cluster do OCP com o back-end do CVO.
2. Revise as instâncias de computação. Isso pode ser um servidor bare metal ou VMs no GCP.

3. Se o tipo de instância ainda não corresponder aos requisitos mínimos de recursos do Astra para nós mestres e trabalhadores, altere o tipo de instância no GCP para atender aos requisitos do Astra. "Requisitos do Astra Control Center"Consulte a .
4. Crie pelo menos um bucket do GCP Cloud Storage para armazenar seus backups.
5. Crie um segredo, que é necessário para o acesso ao bucket.
6. (Opcional) se você não puder acessar o Registro de imagens do NetApp, faça o seguinte:
   a. Crie um Registro de contêiner do Google para hospedar as imagens do Astra Control Center.
   b. Configure o acesso do Google Container Registry para push/pull do Docker para todas as imagens do Astra Control Center.

      Exemplo: As imagens do Astra Control Center podem ser enviadas para esse Registro inserindo o seguinte script:

      ```
      gcloud auth activate-service-account <service account email address>
      --key-file=<GCP Service Account JSON file>
      ```

      Este script requer um arquivo de manifesto Astra Control Center e sua localização do Registro de imagens do Google. Exemplo:

```
manifestfile=acc.manifest.bundle.yaml
GCP_CR_REGISTRY=<target GCP image registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image
    docker push $GCP_CR_REGISTRY/$image
done < acc.manifest.bundle.yaml
```

7. Configurar zonas DNS.

Configurar o NetApp BlueXP para GCP

Usando o NetApp BlueXP , crie uma área de trabalho, adicione um conetor ao GCP, crie um ambiente de trabalho e importe o cluster.

Siga a documentação do BlueXP para concluir as etapas a seguir. "Introdução ao Cloud Volumes ONTAP no GCP"Consulte .

Antes de começar

- Acesso à conta do serviço do GCP com as permissões e funções necessárias do IAM

Passos

1. Adicione suas credenciais ao BlueXP . "Adicionando contas do GCP"Consulte .
2. Adicione um conetor para o GCP.
   a. Escolha "GCP" como Provedor.
   b. Insira as credenciais do GCP. "Criando um conetor no GCP a partir do BlueXP "Consulte .
   c. Certifique-se de que o conetor está a funcionar e mude para esse conetor.
3. Crie um ambiente de trabalho para seu ambiente de nuvem.
   a. Localização: "GCP"
   b. Tipo: "Cloud Volumes ONTAP HA"
4. Importe o cluster OpenShift. O cluster se conetará ao ambiente de trabalho que você acabou de criar.
   a. Veja os detalhes do cluster do NetApp selecionando **K8s** > **Lista de clusters** > **Detalhes do cluster**.
   b. No canto superior direito, observe a versão Astra Control Provisioner.
   c. Observe as classes de storage de cluster do Cloud Volumes ONTAP que mostram "NetApp" como o provisionador.

      Isso importa seu cluster Red Hat OpenShift e atribui a ele uma classe de armazenamento padrão. Você seleciona a classe de armazenamento. O Astra Control Provisioner é instalado automaticamente como parte do processo de importação e descoberta.
5. Observe todos os volumes e volumes persistentes nessa implantação do Cloud Volumes ONTAP.

O Cloud Volumes ONTAP pode operar como um nó único ou em alta disponibilidade (HA). Se a HA estiver ativada, observe o status de HA e o status de implantação de nós em execução no GCP.

Instalar o Astra Control Center no GCP

Siga o padrão "Instruções de instalação do Astra Control Center".

O GCP usa o tipo de bucket Generic S3.

1. Gere o segredo do Docker para extrair imagens para a instalação do Astra Control Center:

```
kubectl create secret docker-registry <secret name> --docker
-server=<Registry location> --docker-username=_json_key --docker
-password="$(cat <GCP Service Account JSON file>)" --namespace=pcloud
```

Implante o Astra Control Center no Microsoft Azure

É possível implantar o Astra Control Center em um cluster Kubernetes autogerenciado, hospedado em uma nuvem pública do Microsoft Azure.

O que você precisará para o Azure

Antes de implantar o Astra Control Center no Azure, você precisará dos seguintes itens:

- Licença do Astra Control Center. "Requisitos de licenciamento do Astra Control Center"Consulte a .

- "Atender aos requisitos do Astra Control Center".
- Conta do NetApp Cloud Central
- Se estiver usando OCP, Red Hat OpenShift Container Platform (OCP) 4,11 a 4,13
- Se estiver usando OCP, permissões do Red Hat OpenShift Container Platform (OCP) (no nível do namespace para criar pods)
- Credenciais do Azure com permissões que permitem criar buckets e conetores

Requisitos de ambiente operacional para o Azure

Certifique-se de que o ambiente operacional escolhido para hospedar o Astra Control Center atenda aos requisitos básicos de recursos descritos na documentação oficial do ambiente.

O Astra Control Center requer recursos específicos, além dos requisitos de recursos do ambiente. "Requisitos do ambiente operacional do Astra Control Center"Consulte a .

Visão geral da implantação para o Azure

Aqui está uma visão geral do processo para instalar o Astra Control Center para Azure.

Cada uma destas etapas é explicada em mais detalhes abaixo.

1. Instale um cluster RedHat OpenShift no Azure.
2. Criar grupos de recursos do Azure.
3. Certifique-se de que tem permissões IAM suficientes.
4. Configurar o Azure.
5. Configure o NetApp BlueXP (anteriormente Gerenciador de nuvem) para Azure.
6. Instalar e configurar o Astra Control Center para Azure.

Instale um cluster RedHat OpenShift no Azure

O primeiro passo é instalar um cluster RedHat OpenShift no Azure.

Para obter instruções de instalação, consulte o seguinte:

- "Instalando o cluster OpenShift no Azure".
- "Instalando uma conta do Azure".

Criar grupos de recursos do Azure

Crie pelo menos um grupo de recursos do Azure.

> OpenShift pode criar seus próprios grupos de recursos. Além disso, você também deve definir grupos de recursos do Azure. Consulte a documentação do OpenShift.

Você pode querer criar um grupo de recursos de cluster de plataforma e um grupo de recursos de cluster OpenShift de aplicativo de destino.

Certifique-se de que tem permissões IAM suficientes

Verifique se você tem funções e permissões suficientes do IAM que permitem instalar um cluster do RedHat

OpenShift e um NetApp BlueXP Connector.

```
https://docs.netapp.com/us-en/cloud-manager-setup-admin/concept-accounts-
azure.html["Credenciais e permissões do Azure"^]Consulte .
```

Configurar o Azure

Em seguida, configure o Azure para criar uma rede virtual, configurar instâncias de computação e criar um contentor Blob do Azure. Se você não puder acessar o Registro de imagem do NetApp, também precisará criar um Registro de contentor do Azure (ACR) para hospedar as imagens do Centro de Controle do Astra e enviar as imagens para esse Registro.

Siga a documentação do Azure para concluir as etapas a seguir. "Instalando o cluster OpenShift no Azure"Consulte .

1. Crie uma rede virtual do Azure.
2. Revise as instâncias de computação. Isso pode ser um servidor bare metal ou VMs no Azure.
3. Se o tipo de instância ainda não corresponder aos requisitos mínimos de recursos do Astra para nós mestres e trabalhadores, altere o tipo de instância no Azure para atender aos requisitos do Astra. "Requisitos do Astra Control Center"Consulte a .
4. Crie pelo menos um contêiner do Blob do Azure para armazenar seus backups.
5. Crie uma conta de armazenamento. Você precisará de uma conta de storage para criar um contêiner para ser usado como um bucket no Astra Control Center.
6. Crie um segredo, que é necessário para o acesso ao bucket.
7. (Opcional) se você não puder acessar o Registro de imagens do NetApp, faça o seguinte:
    a. Crie um ACR (Azure Container Registry) para hospedar as imagens do Astra Control Center.
    b. Configure o acesso ACR para push/pull do Docker para todas as imagens do Astra Control Center.
    c. Envie as imagens do Astra Control Center para esse Registro usando o seguinte script:

        ```
        az acr login -n <AZ ACR URL/Location>
        This script requires the Astra Control Center manifest file and your
        Azure ACR location.
        ```

        Exemplo:

```
manifestfile=acc.manifest.bundle.yaml
AZ_ACR_REGISTRY=<target Azure ACR image registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image
    docker push $AZ_ACR_REGISTRY/$image
done < acc.manifest.bundle.yaml
```

8. Configurar zonas DNS.

Configure o NetApp BlueXP (anteriormente Gerenciador de nuvem) para Azure

Usando o BlueXP (antigo Gerenciador de nuvem), crie uma área de trabalho, adicione um conetor ao Azure, crie um ambiente de trabalho e importe o cluster.

Siga a documentação do BlueXP para concluir as etapas a seguir. "Introdução ao BlueXP no Azure"Consulte .

Antes de começar

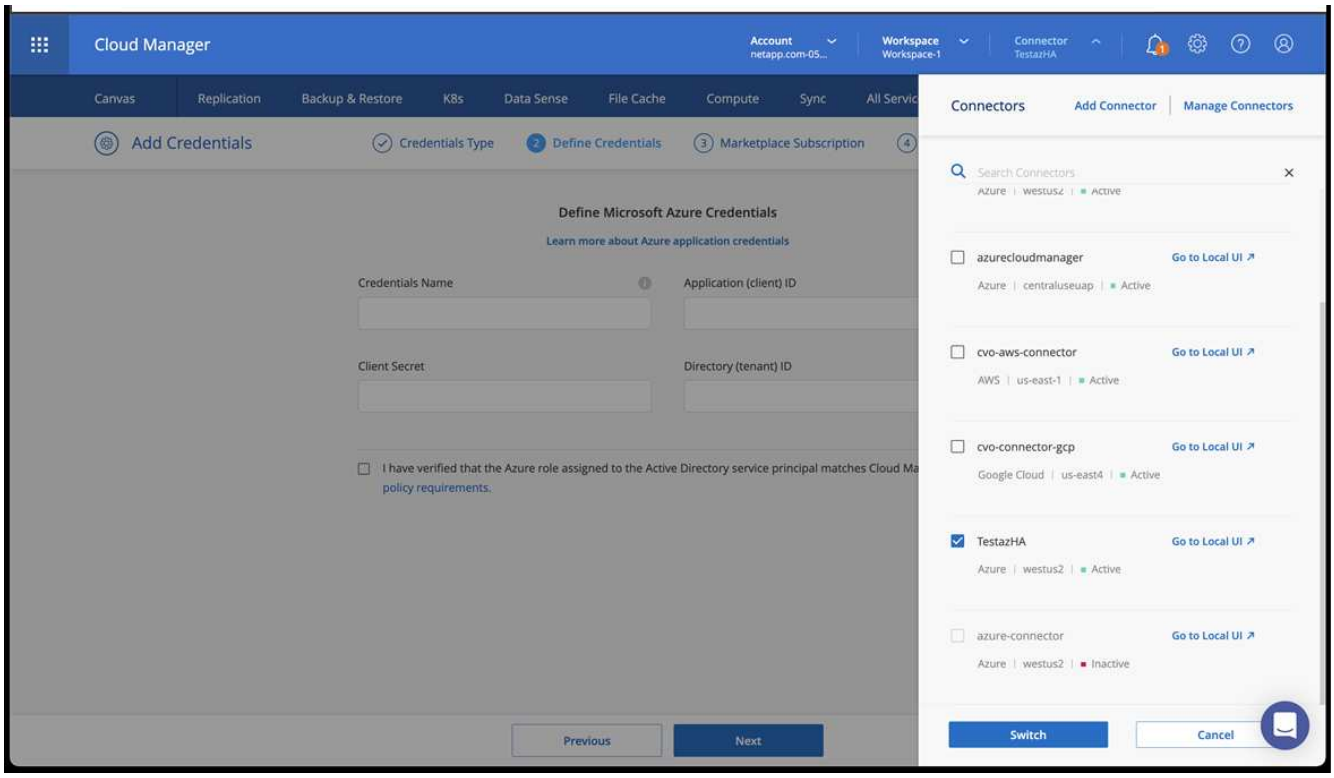
Acesso à conta do Azure com as permissões e funções necessárias do IAM

Passos

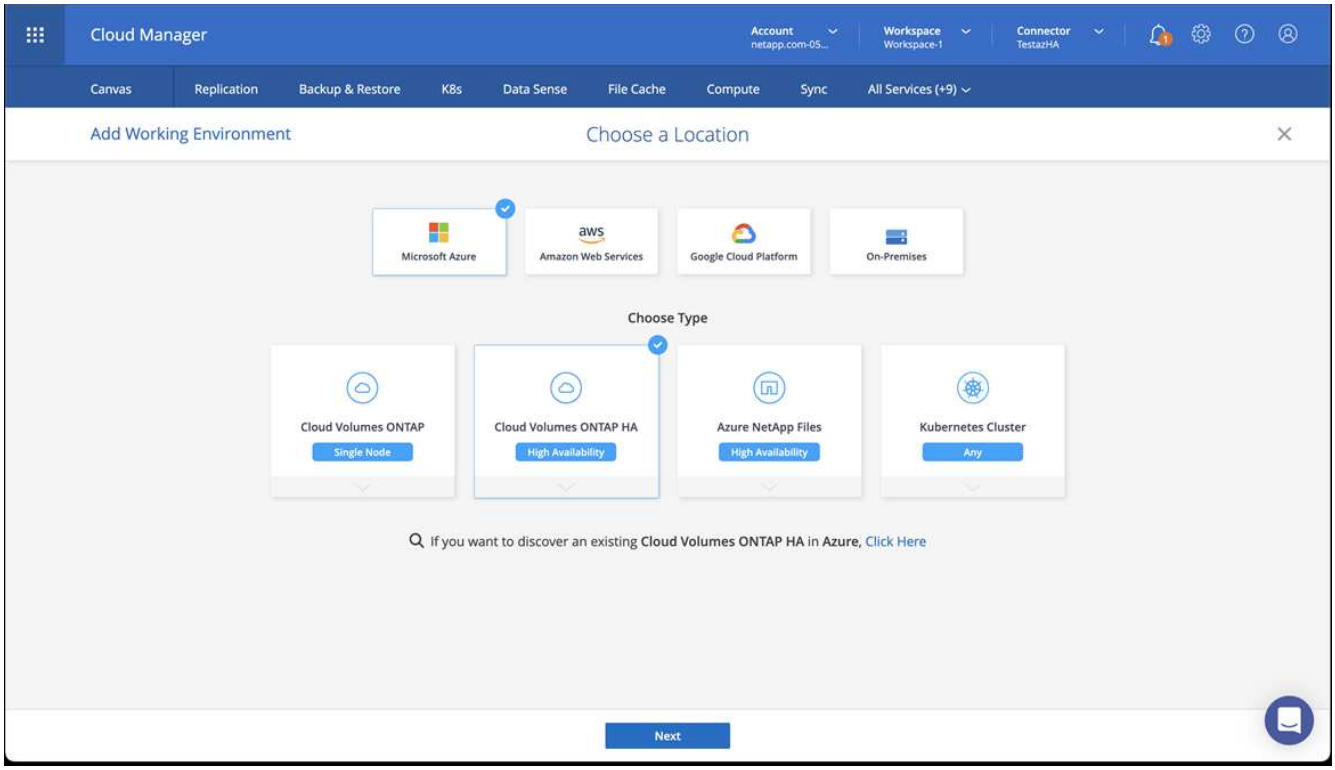
1. Adicione suas credenciais ao BlueXP .
2. Adicione um conetor para o Azure. "Políticas da BlueXP "Consulte .
   a. Escolha **Azure** como Provedor.
   b. Insira as credenciais do Azure, incluindo o ID do aplicativo, o segredo do cliente e o ID do diretório (locatário).

   ```
   https://docs.netapp.com/us-
   en/occm/task_creating_connectors_azure.html["Criando um conetor no
   Azure a partir do BlueXPr"^]Consulte .
   ```

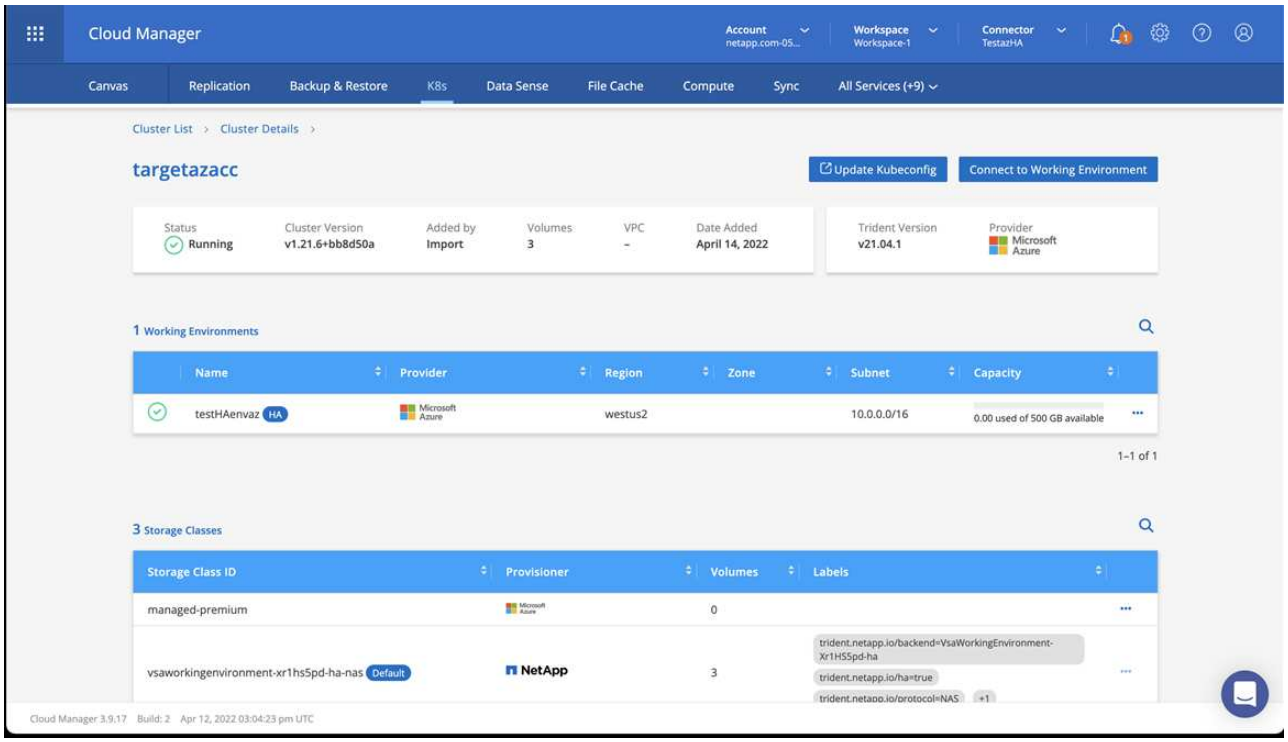
3. Certifique-se de que o conetor está a funcionar e mude para esse conetor.



4. Crie um ambiente de trabalho para seu ambiente de nuvem.
   a. Localização: "Microsoft Azure".
   b. Tipo: "Cloud Volumes ONTAP HA".



5. Importe o cluster OpenShift. O cluster se conetará ao ambiente de trabalho que você acabou de criar.
   a. Veja os detalhes do cluster do NetApp selecionando **K8s** > **Lista de clusters** > **Detalhes do cluster**.



b. No canto superior direito, observe a versão Astra Control Provisioner.

c. Observe as classes de storage de cluster do Cloud Volumes ONTAP que mostram o NetApp como o provisionador.

Isso importa seu cluster Red Hat OpenShift e atribui uma classe de armazenamento padrão. Você seleciona a classe de armazenamento. O Astra Control Provisioner é instalado automaticamente como parte do processo de importação e descoberta.

6. Observe todos os volumes e volumes persistentes nessa implantação do Cloud Volumes ONTAP.
7. O Cloud Volumes ONTAP pode operar como um único nó ou em alta disponibilidade. Se a HA estiver ativada, observe o status da HA e o status da implantação do nó em execução no Azure.

Instalar e configurar o Astra Control Center para Azure

Instalar o Astra Control Center com o padrão "instruções de instalação".

Usando o Astra Control Center, adicione um bucket do Azure. "Configure o Astra Control Center e adicione buckets"Consulte a .

Configure o Astra Control Center após a instalação

Dependendo do seu ambiente, pode haver configuração adicional necessária após a instalação do Astra Control Center.

Remover limitações de recursos

Alguns ambientes usam os objetos ResourceQuotes e LimitRanges para impedir que os recursos em um namespace consumam toda a CPU e memória disponíveis no cluster. O Astra Control Center não define limites máximos, por isso não estará em conformidade com esses recursos. Se o seu ambiente estiver configurado dessa forma, você precisará remover esses recursos dos namespaces onde você planeja instalar o Astra Control Center.

Você pode usar as etapas a seguir para recuperar e remover essas cotas e limites. Nestes exemplos, a saída do comando é mostrada imediatamente após o comando.

Passos

1. Obtenha as cotas de recursos no `netapp-acc` namespace (ou nome personalizado):

```
kubectl get quota -n [netapp-acc or custom namespace]
```

Resposta:

```
NAME          AGE   REQUEST                                                 LIMIT
pods-high     16s   requests.cpu: 0/20, requests.memory: 0/100Gi
limits.cpu: 0/200, limits.memory: 0/1000Gi
pods-low      15s   requests.cpu: 0/1, requests.memory: 0/1Gi
limits.cpu: 0/2, limits.memory: 0/2Gi
pods-medium   16s   requests.cpu: 0/10, requests.memory: 0/20Gi
limits.cpu: 0/20, limits.memory: 0/200Gi
```

2. Excluir todas as cotas de recursos por nome:

```
kubectl delete resourcequota pods-high -n [netapp-acc or custom
namespace]
```

```
kubectl delete resourcequota pods-low -n [netapp-acc or custom
namespace]
```

```
kubectl delete resourcequota pods-medium -n [netapp-acc or custom
namespace]
```

3. Obtenha os intervalos de limite no `netapp-acc` namespace (ou nome personalizado):

```
kubectl get limits -n [netapp-acc or custom namespace]
```

Resposta:

```
NAME              CREATED AT
cpu-limit-range   2022-06-27T19:01:23Z
```

4. Eliminar os intervalos de limite por nome:

```
kubectl delete limitrange cpu-limit-range -n [netapp-acc or custom
namespace]
```

Adicione um certificado TLS personalizado

O Astra Control Center usa um certificado TLS autoassinado por padrão para o tráfego do controlador de entrada (somente em certas configurações) e autenticação da IU da Web com navegadores da Web. Para uso em produção, você deve remover o certificado TLS autoassinado existente e substituí-lo por um certificado TLS assinado por uma Autoridade de Certificação (CA).

> ⓘ O certificado auto-assinado padrão é usado para dois tipos de conexões:
>
> - Conexões HTTPS com a IU da Web do Astra Control Center
> - Tráfego do controlador de entrada (somente se a `ingressType: "AccTraefik"` propriedade foi definida no `astra_control_center.yaml` arquivo durante a instalação do Astra Control Center)
>
> A substituição do certificado TLS padrão substitui o certificado usado para autenticação dessas conexões.

Antes de começar

- Cluster do Kubernetes com Astra Control Center instalado
- Acesso administrativo a um shell de comando no cluster para executar `kubectl` comandos
- Arquivos de chave privada e certificado da CA

Remova o certificado autoassinado

Remova o certificado TLS autoassinado existente.

1. Usando SSH, faça login no cluster do Kubernetes que hospeda o Astra Control Center como usuário administrativo.
2. Localize o segredo TLS associado ao certificado atual usando o seguinte comando, substituindo `<ACC-deployment-namespace>` pelo namespace de implantação do Astra Control Center:

```
kubectl get certificate -n <ACC-deployment-namespace>
```

3. Exclua o segredo e o certificado atualmente instalados usando os seguintes comandos:

```
kubectl delete cert cert-manager-certificates -n <ACC-deployment-
namespace>
```

```
kubectl delete secret secure-testing-cert -n <ACC-deployment-namespace>
```

Adicione um novo certificado usando a linha de comando

Adicione um novo certificado TLS assinado por uma CA.

1. Use o comando a seguir para criar o novo segredo TLS com a chave privada e os arquivos de certificado da CA, substituindo os argumentos entre colchetes> pelas informações apropriadas:

```
kubectl create secret tls <secret-name> --key <private-key-filename>
--cert <certificate-filename> -n <ACC-deployment-namespace>
```

2. Use o comando e exemplo a seguir para editar o arquivo CRD (Custom Resource Definition) do cluster e altere o `spec.selfSigned` valor para `spec.ca.secretName` se referir ao segredo TLS criado anteriormente:

```
kubectl edit clusterissuers.cert-manager.io/cert-manager-certificates -n
<ACC-deployment-namespace>
```

CRD:

```
#spec:
#  selfSigned: {}

spec:
  ca:
    secretName: <secret-name>
```

3. Use o comando e exemplo de saída a seguir para validar se as alterações estão corretas e o cluster está pronto para validar certificados, substituindo `<ACC-deployment-namespace>` pelo namespace de implantação do Astra Control Center:

```
kubectl describe clusterissuers.cert-manager.io/cert-manager-
certificates -n <ACC-deployment-namespace>
```

Resposta:

```
Status:
  Conditions:
    Last Transition Time:  2021-07-01T23:50:27Z
    Message:               Signing CA verified
    Reason:                KeyPairVerified
    Status:                True
    Type:                  Ready
Events:                    <none>
```

4. Crie o `certificate.yaml` arquivo usando o exemplo a seguir, substituindo os valores de espaço reservado entre colchetes> por informações apropriadas:

> Este exemplo usa a propriedade a `dnsNames` para especificar o endereço DNS do Astra Control Center. O Astra Control Center não oferece suporte ao uso da propriedade Nome Comum (CN) para especificar o endereço DNS.

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  <strong>name: <certificate-name></strong>
  namespace: <ACC-deployment-namespace>
spec:
  <strong>secretName: <certificate-secret-name></strong>
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  dnsNames:
  <strong>- <astra.dnsname.example.com></strong> #Replace with the
correct Astra Control Center DNS address
  issuerRef:
    kind: ClusterIssuer
    name: cert-manager-certificates
```

5. Crie o certificado usando o seguinte comando:

```
kubectl apply -f certificate.yaml
```

6. Usando o comando a seguir e exemplo de saída, valide que o certificado foi criado corretamente e com os argumentos especificados durante a criação (como nome, duração, prazo de renovação e nomes DNS).

```
kubectl describe certificate -n <ACC-deployment-namespace>
```

Resposta:

```
Spec:
  Dns Names:
    astra.example.com
  Duration:  125h0m0s
  Issuer Ref:
    Kind:        ClusterIssuer
    Name:        cert-manager-certificates
  Renew Before:  61h0m0s
  Secret Name:   <certificate-secret-name>
Status:
  Conditions:
    Last Transition Time:  2021-07-02T00:45:41Z
    Message:               Certificate is up to date and has not expired
    Reason:                Ready
    Status:                True
    Type:                  Ready
  Not After:               2021-07-07T05:45:41Z
  Not Before:              2021-07-02T00:45:41Z
  Renewal Time:            2021-07-04T16:45:41Z
  Revision:                1
Events:                    <none>
```

7. Edite o TLS armazena o CRD para apontar para o novo nome secreto do certificado usando o comando e o exemplo a seguir, substituindo os valores do espaço reservado entre parênteses> por informações apropriadas

```
kubectl edit tlsstores.traefik.io -n <ACC-deployment-namespace>
```

CRD:

```
...
spec:
  defaultCertificate:
    secretName: <certificate-secret-name>
```

8. Edite a opção TLS de CRD de entrada para apontar para o novo segredo de certificado usando o comando e o exemplo a seguir, substituindo os valores de espaço reservado entre colchetes> por informações apropriadas:

```
kubectl edit ingressroutes.traefik.io -n <ACC-deployment-namespace>
```

CRD:

```
...
 tls:
    secretName: <certificate-secret-name>
```

9. Usando um navegador da Web, navegue até o endereço IP de implantação do Astra Control Center.
10. Verifique se os detalhes do certificado correspondem aos detalhes do certificado que você instalou.
11. Exporte o certificado e importe o resultado para o gerenciador de certificados no navegador da Web.

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:

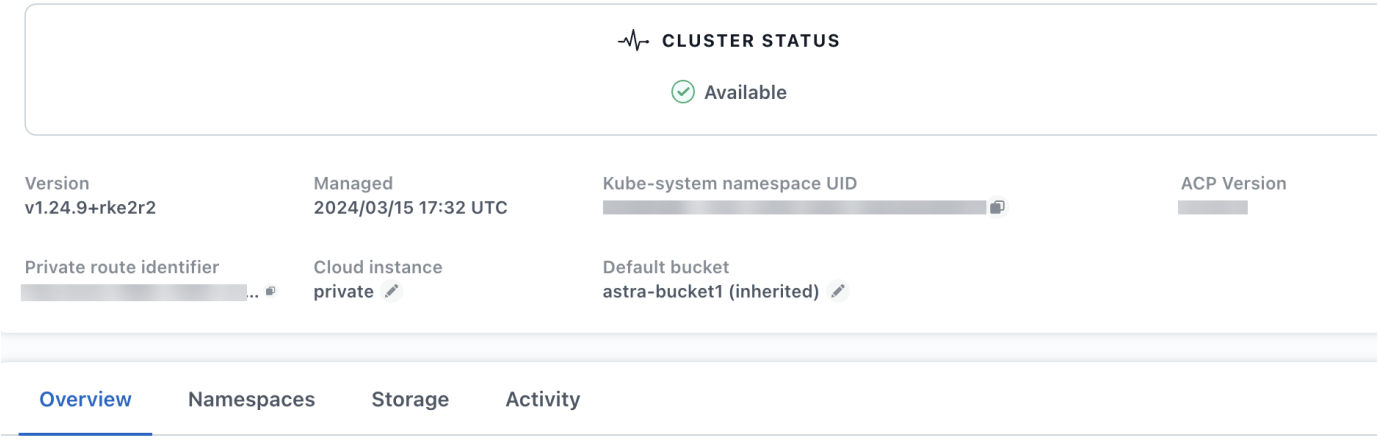
```
./tridentctl uninstall -n trident
```

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