



Visão geral da instalação

Astra Control Center

NetApp
August 11, 2025

Índice

Visão geral da instalação	1
Instale o Astra Control Center usando o processo padrão	1
Faça download e extraia Astra Control Center	2
Instale o plug-in NetApp Astra kubectl	3
Adicione as imagens ao seu registro local	3
Configure namespace e segredo para Registros com requisitos de autenticação	5
Instale o operador do Centro de Controle Astra	7
Configurar o Astra Control Center	10
Instalação completa do operador e do Centro de Controle Astra	24
Verifique o status do sistema	25
Configure a entrada para o balanceamento de carga	30
Faça login na IU do Astra Control Center	34
Solucionar problemas da instalação	34
O que vem a seguir	35
Configurar um gerenciador de cert externo	35
Instale o Astra Control Center usando o OpenShift OperatorHub	37
Faça download e extraia Astra Control Center	38
Instale o plug-in NetApp Astra kubectl	38
Adicione as imagens ao seu registro local	39
Localize a página de instalação do operador	41
Instale o operador	42
Instale o Astra Control Center	43
Crie um segredo de Registro	44
O que vem a seguir	45
Instalar o Astra Control Center com um back-end de storage do Cloud Volumes ONTAP	45
Implante o Astra Control Center na Amazon Web Services	46
Implante o Astra Control Center no Google Cloud Platform	50
Implante o Astra Control Center no Microsoft Azure	54
Configure o Astra Control Center após a instalação	60
Remover limitações de recursos	60
Ative a comunicação de rede entre namespaces	62
Adicione um certificado TLS personalizado	63

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, faça o download do pacote de instalação no site de suporte da NetApp e execute as etapas a seguir. Você pode usar este procedimento para instalar o Astra Control Center em ambientes conectados à Internet ou com conexão via rede.

Outros procedimentos de instalação

- **Instalar com RedHat 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.

Para uma demonstração do processo de instalação do Astra Control Center, ["este vídeo"](#) consulte .

Antes de começar

- ["Antes de começar a instalação, prepare seu ambiente para a implantação do Astra Control Center"](#).
- Se você configurou ou deseja configurar políticas de segurança de pod no seu ambiente, familiarize-se com as políticas de segurança de pod e como elas afetam a instalação do Astra Control Center. ["Compreender as restrições da política de segurança do pod"](#) Consulte a .
- Certifique-se de que todos os serviços de API estão em um estado saudável e disponíveis:

```
kubectl get apiservices
```

- Certifique-se de que o Astra FQDN que pretende utilizar está roteável para este 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.
- 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.



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.

Sobre esta tarefa

O processo de instalação do Astra Control Center ajuda você a fazer o seguinte:

- Instale os componentes do Astra no `netapp-acc` namespace (ou nome personalizado).
- Crie uma conta de administrador padrão do Astra Control Owner.
- Estabeleça um endereço de e-mail de usuário administrativo e uma senha de configuração inicial padrão. Este usuário é atribuído a função proprietário que é necessária para o primeiro login na IU.
- Determine se todos os pods do Astra Control Center estão em execução.
- Instale a 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.

Passos

Para instalar o Astra Control Center, siga estas etapas:

- [Faça download e extraia Astra Control Center](#)
- [Instale o plug-in NetApp Astra kubectl](#)
- [Adicione as imagens ao seu registo 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](#)

Faça download e extraia Astra Control Center

1. Vá para o ["Página de downloads do Astra Control Center"](#) no site de suporte da NetApp.
2. Faça o download do pacote que contém o Astra Control Center (`astra-control-center-[version].tar.gz`).
3. (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.

4. Extraia as imagens do pacote Astra Control Center:

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

Instale o plug-in NetApp Astra kubectl

Você pode usar o plugin de linha de comando NetApp Astra kubectl para enviar imagens para um repositório local do Docker.

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 local

1. Complete a sequência de passos adequada para o seu motor de contentores:

Docker

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

```
acc.manifest.bundle.yaml
acc/
```

2. 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>\"</a>`, .
- Substitua o `<MY_REGISTRY_USER>` pelo nome de usuário.
- Substitua o `<MY_REGISTRY_TOKEN>` por um token autorizado para o Registro.

```
kubect1 astra packages push-images -m <BUNDLE_FILE> -r
<MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
<MY_REGISTRY_TOKEN>
```

Podman

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

```
acc.manifest.bundle.yaml
acc/
```

2. Inicie sessão no seu registro:

```
podman login <YOUR_REGISTRY>
```

3. 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=23.04.2-7
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

```

Podman 3

```

export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.04.2-7
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://netappdownloads.jfrog.io/docker-astra-control-
prod/netapp/astra/acc/23.04.2-7/image:version

```

Configure namespace e segredo para Registros com requisitos de autenticação

1. Exporte o KUBECONFIG para o cluster host do Astra Control Center:

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



Antes de concluir a instalação, certifique-se de que o KUBECONFIG está apontando para o cluster onde você deseja instalar o Astra Control Center. O KUBECONFIG pode conter apenas um contexto.

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
```

Resposta:

```
namespace/netapp-acc-operator created
```

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/23.04.2-7`).

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

Resposta da amostra:

```
secret/astra-registry-cred created
```



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

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

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

Resposta da amostra:


```
namespace/netapp-acc created
```

- d. Crie um segredo para o netapp-acc namespace (ou nome personalizado). Adicione informações do Docker e execute o seguinte comando:

```
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]
```

Resposta

```
secret/astra-registry-cred created
```

Instale o operador do Centro de Controle Astra

1. Altere o diretório:

```
cd manifests
```

2. Edite a implantação do operador Astra Control Center YAML)
(`astra\_control\_center\_operator\_deploy.yaml` para consultar o Registro local e o segredo.

```
vim astra_control_center_operator_deploy.yaml
```



Uma amostra anotada YAML segue estes passos.

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

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

- b. Altere `[your_registry_path]` para a `kube-rbac-proxy` imagem para o caminho do registro onde as imagens foram empurradas para um [passo anterior](#).
- c. Altere `[your_registry_path]` para a `acc-operator-controller-manager` imagem para o caminho do registro onde as imagens foram empurradas para um [passo anterior](#).

```
<strong>astra_control_center_operator_deploy.yaml</strong>
```

```
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: [your_registry_path]/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: [your_registry_path]/acc-operator:23.04.36
          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

```

3. Instale o operador do Centro de Controle Astra:

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

Resposta da amostra:

```

namespace/netapp-acc-operator created
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.astra.
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

```

4. 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:

`<code>accountName</code>`

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

`<code>astraVersion</code>`

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 caracteres	23.04.2-7

<code> </code>

Definição	Orientação	Tipo	Exemplo
astraAddress	Altere a astraAddress cadeia de caracteres 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 caracteres	astra.example.com

<code> AutoSupport </code>

Suas seleções nesta seção determinam se você participará do aplicativo de suporte Pro-ativo da NetApp, NetApp Active IQ 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
<code>autoSupport.enrolled</code>	enrolled`Os campos ou `url têm de ser selecionados	Alterar enrolled para AutoSupport para false sites sem conectividade com a Internet ou manter true para sites conectados. 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)
<code>autoSupport.url</code>	enrolled`Os campos ou `url têm de ser selecionados	Esta URL determina onde os dados anônimos serão enviados.	cadeia de caracteres	https://support.netapp.com/asupprod/post/1.0/postAsup

`<code> email</code>`

Definição	Orientação	Tipo	Exemplo
email	Altere a email cadeia de caracteres 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 caracteres	admin@example.com

`<code>firstName</code>`

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 caracteres	SRE

`<code>LastName</code>`

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 caracteres	Admin

<code> imageRegistry</code>

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
<code>imageRegistry.name</code>	Obrigatório	O nome do registo de imagens onde as imagens foram enviadas para o passo anterior . Não utilize <code>http://</code> ou <code>https://</code> no nome do registo.	cadeia de caracteres	<code>example.registry.com/astra</code>
<code>imageRegistry.secret</code>	Obrigatório se a cadeia de caracteres inserida para <code>imageRegistry.name</code> requires a secret. IMPORTANT: If you are using a registry that does not require authorization, you must delete this <code>`secret</code> a linha <code>imageRegistry</code> ou a instalação falhar.	O nome do segredo do Kubernetes usado para autenticar com o Registro de imagens.	cadeia de caracteres	<code>astra-registry-cred</code>

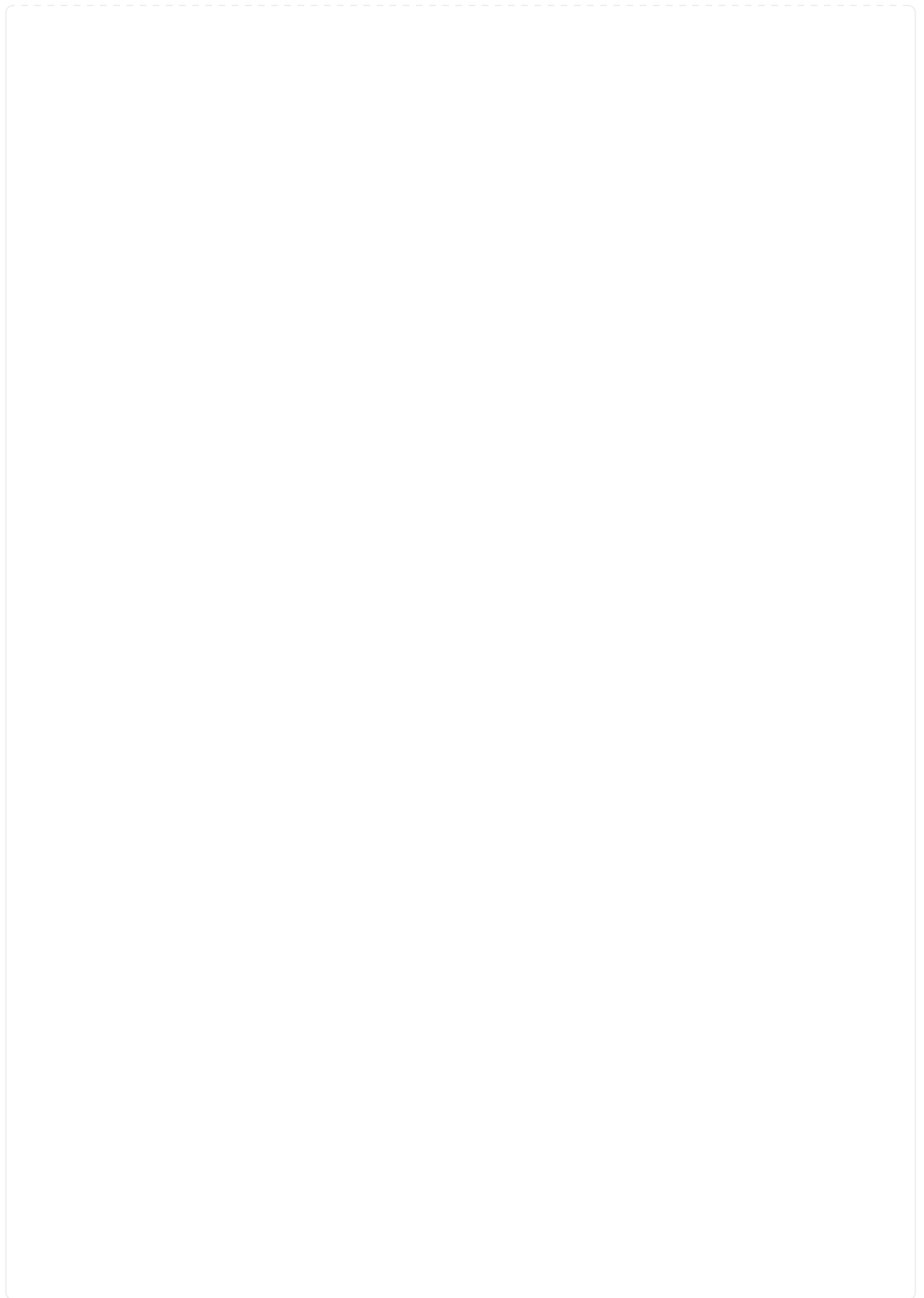
`<code>storageClass</code>`

Definição	Orientação	Tipo	Exemplo
storageClass	Altere storageClass o valor de ontap-gold para outro recurso de storageClass do Astra Trident, conforme exigido pela sua instalação. Execute o comando <code>kubectl get sc</code> para determinar suas classes de armazenamento configuradas existentes. Uma das classes de storage baseadas no Astra Trident deve ser inserida no arquivo MANIFEST (<code>astra-control-center-<version>.manifest</code>) 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 caracteres	ontap-gold

`<code> volume ReclaimPolicy</code>`

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

`<code>ingressType</code>`



Definição	Orientação	Tipo	Opções
ingressType	Use um dos seguintes tipos de entrada: Generic (ingressType: "Generic") (Padrão) Use esta opção quando tiver outro controlador de entrada em uso ou preferir usar 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 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 externo para atribuir um endereço IP externo ao serviço. Na configuração do servidor DNS interno, você deve apontar o nome DNS escolhido	cadeia de caracteres	• Generic (este é o valor padrão) • AccTraefik

`<code> scaleSize</code>`

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 caracteres	• Small • Medium (Este é o valor padrão)

<code>astraResourcesScaler</code>

Definição	Orientação	Tipo	Opções
<code>astraResourcesScaler</code>	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 <code>CR</code> <code>astraResourcesScaler</code> pode ser definido como <code>Off</code>. Isso desativa as solicitações de recursos e permite a implantação em clusters menores.	cadeia de caracteres	• <code>Default</code> (Este é o valor padrão) • <code>Off</code>

`<code> AdditionalValues</code>`

- Para a comunicação Astral Control Center e Cloud Insights, a verificação de certificado TLS é desativada por padrão. Você pode habilitar a verificação de certificação TLS para comunicação entre o Cloud Insights e o cluster de host e o cluster gerenciado do Astra Control Center adicionando a seguinte seção em `additionalValues`.

```
additionalValues:
  netapp-monitoring-operator:
    config:
      ciSkipTlsVerify: false
  cloud-insights-service:
    config:
      ciSkipTlsVerify: false
  telemetry-service:
    config:
      ciSkipTlsVerify: false
```


`<code> crds</code>`

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

Definição	Orientação	Tipo	Exemplo
<code>crds.externalCertManager</code>	Se você usar um gerenciador cert externo, <code>externalCertManager</code> altere para <code>true</code> . O padrão <code>false</code> 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	<code>False</code> (este valor é o padrão)
<code>crds.externalTraefik</code>	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	<code>False</code> (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.

```
<strong>astra_control_center.yaml</strong>
```

```
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: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  scaleSize: "Medium"
  astraResourcesScaler: "Default"
  additionalValues: {}
  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]
```

Resposta da amostra:

```
namespace/netapp-acc created
```

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

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

Resposta da amostra:

```
astracontrolcenter.astra.netapp.io/astra created
```



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 seção](#) Consulte 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.

Resposta da amostra

NAME	READY	STATUS	
RESTARTS AGE			
acc-helm-repo-6cc7696d8f-pmhm8 9h	1/1	Running	0
activity-597fb656dc-5rd4l 9h	1/1	Running	0
activity-597fb656dc-mqmcw 9h	1/1	Running	0
api-token-authentication-62f84 9h	1/1	Running	0
api-token-authentication-68nlf 9h	1/1	Running	0
api-token-authentication-ztgrm 9h	1/1	Running	0
asup-669d4ddbc4-fnmwp (9h ago) 9h	1/1	Running	1
authentication-78789d7549-lk686 9h	1/1	Running	0
bucket-service-65c7d95496-24x7l (9h ago) 9h	1/1	Running	3
cert-manager-c9f9fbf9f-k8zq2 9h	1/1	Running	0
cert-manager-c9f9fbf9f-qj1zm 9h	1/1	Running	0
cert-manager-cainjector-dbbbd8447-b5ql1 9h	1/1	Running	0
cert-manager-cainjector-dbbbd8447-p5whs 9h	1/1	Running	0
cert-manager-webhook-6f97bb7d84-4722b 9h	1/1	Running	0
cert-manager-webhook-6f97bb7d84-86kv5 9h	1/1	Running	0
certificates-59d9f6f4bd-2j899 9h	1/1	Running	0
certificates-59d9f6f4bd-9d9k6 9h	1/1	Running	0
certificates-expiry-check-28011180--1-8lkxz 9h	0/1	Completed	0
cloud-extension-5c9c9958f8-jdhrp 9h	1/1	Running	0
cloud-insights-service-5cdd5f7f-pp8r5 9h	1/1	Running	0
composite-compute-66585789f4-hxn5w 9h	1/1	Running	0

composite-volume-68649f68fd-tb7p4 9h	1/1	Running	0
credentials-dfc844c57-jsx92 9h	1/1	Running	0
credentials-dfc844c57-xw26s 9h	1/1	Running	0
entitlement-7b47769b87-4jb6c 9h	1/1	Running	0
features-854d8444cc-c24b7 9h	1/1	Running	0
features-854d8444cc-dv6sm 9h	1/1	Running	0
fluent-bit-ds-9tlv4 9h	1/1	Running	0
fluent-bit-ds-bpkcb 9h	1/1	Running	0
fluent-bit-ds-cxmwx 9h	1/1	Running	0
fluent-bit-ds-jgnhc 9h	1/1	Running	0
fluent-bit-ds-vtr6k 9h	1/1	Running	0
fluent-bit-ds-vxqd5 9h	1/1	Running	0
graphql-server-7d4b9d44d5-zdbf5 9h	1/1	Running	0
identity-6655c48769-4pwk8 9h	1/1	Running	0
influxdb2-0 9h	1/1	Running	0
keycloak-operator-55479d6fc6-slvmt 9h	1/1	Running	0
krakend-f487cb465-78679 9h	1/1	Running	0
krakend-f487cb465-rjsxx 9h	1/1	Running	0
license-64cbc7cd9c-qxsr8 9h	1/1	Running	0
login-ui-5db89b5589-ndb96 9h	1/1	Running	0
loki-0 9h	1/1	Running	0
metrics-facade-8446f64c94-x8h7b 9h	1/1	Running	0
monitoring-operator-6b44586965-pvcl4 9h	2/2	Running	0

nats-0 9h	1/1	Running	0
nats-1 9h	1/1	Running	0
nats-2 9h	1/1	Running	0
nautilus-85754d87d7-756qb 9h	1/1	Running	0
nautilus-85754d87d7-q8j7d 9h	1/1	Running	0
openapi-5f9cc76544-7fnjm 9h	1/1	Running	0
openapi-5f9cc76544-vzr7b 9h	1/1	Running	0
packages-5db49f8b5-lrzhd 9h	1/1	Running	0
polaris-consul-consul-server-0 9h	1/1	Running	0
polaris-consul-consul-server-1 9h	1/1	Running	0
polaris-consul-consul-server-2 9h	1/1	Running	0
polaris-keycloak-0 (9h ago) 9h	1/1	Running	2
polaris-keycloak-1 9h	1/1	Running	0
polaris-keycloak-2 9h	1/1	Running	0
polaris-keycloak-db-0 9h	1/1	Running	0
polaris-keycloak-db-1 9h	1/1	Running	0
polaris-keycloak-db-2 9h	1/1	Running	0
polaris-mongodb-0 9h	1/1	Running	0
polaris-mongodb-1 9h	1/1	Running	0
polaris-mongodb-2 9h	1/1	Running	0
polaris-ui-66fb99479-qp9gq 9h	1/1	Running	0
polaris-vault-0 9h	1/1	Running	0
polaris-vault-1 9h	1/1	Running	0

polaris-vault-2 9h	1/1	Running	0
public-metrics-76fbf9594d-zmxzw 9h	1/1	Running	0
storage-backend-metrics-7d7fbc9cb9-lmd25 9h	1/1	Running	0
storage-provider-5bdd456c4b-2fftc 9h	1/1	Running	0
task-service-87575df85-dnn2q (9h ago) 9h	1/1	Running	3
task-service-task-purge-28011720--1-q6w4r 28m	0/1	Completed	0
task-service-task-purge-28011735--1-vk6pd 13m	1/1	Running	0
telegraf-ds-2r2kw 9h	1/1	Running	0
telegraf-ds-6s9d5 9h	1/1	Running	0
telegraf-ds-96jl7 9h	1/1	Running	0
telegraf-ds-hbp84 9h	1/1	Running	0
telegraf-ds-plwzv 9h	1/1	Running	0
telegraf-ds-sr22c 9h	1/1	Running	0
telegraf-rs-4sbg8 9h	1/1	Running	0
telemetry-service-fb9559f7b-mk9l7 (9h ago) 9h	1/1	Running	3
tenancy-559bbc6b48-5msgg 9h	1/1	Running	0
traefik-d997b8877-7xpf4 9h	1/1	Running	0
traefik-d997b8877-9xv96 9h	1/1	Running	0
trident-svc-585c97548c-d25z5 9h	1/1	Running	0
vault-controller-88484b454-2d6sr 9h	1/1	Running	0
vault-controller-88484b454-fc5cz 9h	1/1	Running	0
vault-controller-88484b454-jktld 9h	1/1	Running	0

4. (Opcional) para garantir que a instalação esteja concluída, você pode assistir os `acc-operator` logs usando o seguinte comando.

```
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	23.04.2-7	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 os seguintes tipos de controlador de entrada:

- Entrada de Istio
- Controlador de entrada nginx

- Controlador de entrada OpenShift

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 `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 registros. 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
```

O que vem a seguir

- (Opcional) dependendo do seu ambiente, conclua a pós-instalação "[etapas de configuração](#)".
- Conclua a implantação executando "[tarefas de configuração](#)"o .

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

- * Pré-requisitos ambientais atendidos*: ["Antes de começar a instalação, prepare seu ambiente para a implantação do Astra Control Center"](#).
- * 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
```

- **Endereço FQDN:** Obtenha um endereço FQDN para o Astra Control Center em seu data center.
- **Permissões OpenShift:** Obtenha as permissões necessárias e o acesso à Red Hat OpenShift Container Platform para executar as etapas de instalação descritas.
- **Gerenciador de cert configurado:** Se um gerenciador de cert já existir no cluster, você precisará executar alguns ["etapas de pré-requisito"](#) para que o Astra Control Center não instale seu próprio gerenciador de cert. Por padrão, o Astra Control Center instala seu próprio gerenciador de cert durante a instalação.
- * Controlador de entrada de Kubernetes*: Se você tiver uma controladora de entrada de 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.

Passos

- [Faça download e extraia Astra Control Center](#)

- [Instale o plug-in NetApp Astra kubectl](#)
- [Adicione as imagens ao seu registo local](#)
- [Localize a página de instalação do operador](#)
- [Instale o operador](#)
- [Instale o Astra Control Center](#)

Faça download e extraia Astra Control Center

1. Vá para o "[Página de downloads do Astra Control Center](#)" no site de suporte da NetApp.
2. Faça o download do pacote que contém o Astra Control Center (`astra-control-center-[version].tar.gz`).
3. (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.

4. Extraia as imagens do pacote Astra Control Center:

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

Instale o plug-in NetApp Astra kubectl

Você pode usar o plugin de linha de comando NetApp Astra kubectl para enviar imagens para um repositório local do Docker.

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.

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 kubect1-astra/
```

2. Mova o binário correto para o caminho atual e renomeie-o para kubect1-astra:

```
cp kubect1-astra/<binary-name> /usr/local/bin/kubect1-astra
```

Adicione as imagens ao seu registo local

1. Complete a sequência de passos adequada para o seu motor de contentores:

Docker

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

```
acc.manifest.bundle.yaml
acc/
```

2. 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>\"</a>`, .
- Substitua o `<MY_REGISTRY_USER>` pelo nome de usuário.
- Substitua o `<MY_REGISTRY_TOKEN>` por um token autorizado para o Registro.

```
kubect1 astra packages push-images -m <BUNDLE_FILE> -r
<MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
<MY_REGISTRY_TOKEN>
```

Podman

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

```
acc.manifest.bundle.yaml
acc/
```

2. Inicie sessão no seu registro:

```
podman login <YOUR_REGISTRY>
```

3. 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=23.04.2-7
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

```

Podman 3

```

export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.04.2-7
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:

```

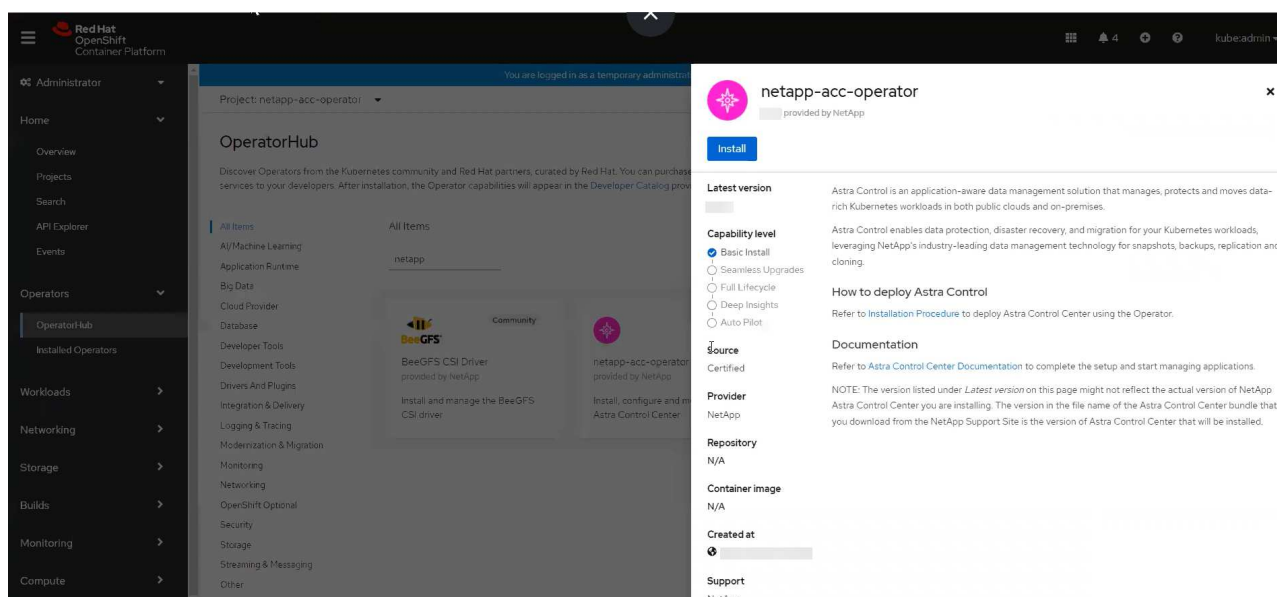
https://netappdownloads.jfrog.io/docker-astra-control-
prod/netapp/astra/acc/23.04.2-7/image:version

```

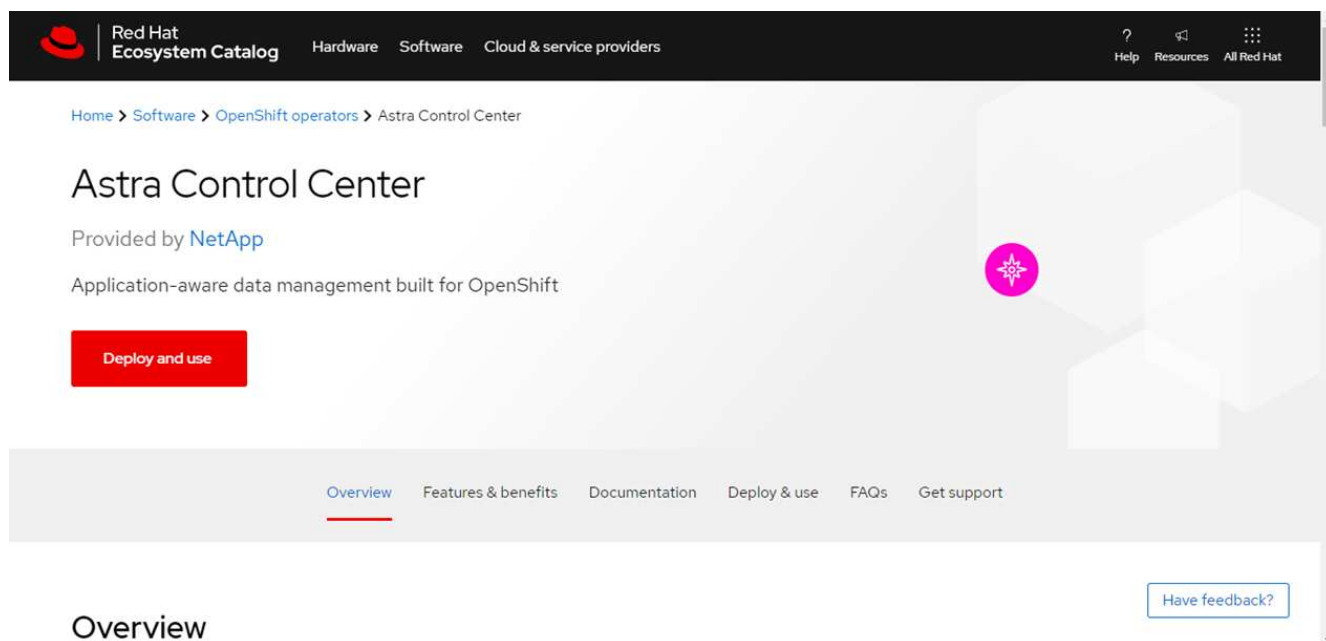
Localize a página de instalação do operador

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

- A partir do console web Red Hat OpenShift:
 - i. Faça login na IU da OpenShift Container Platform.
 - ii. No menu lateral, selecione **operadores > OperatorHub**.
 - iii. Procure e selecione o operador do Centro de Controle NetApp Astra.



- No Red Hat Ecosystem Catalog:
 - i. Selecione o Centro de Controle NetApp Astra "**operador**".
 - ii. 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.

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

- Selecione **Instalar**.

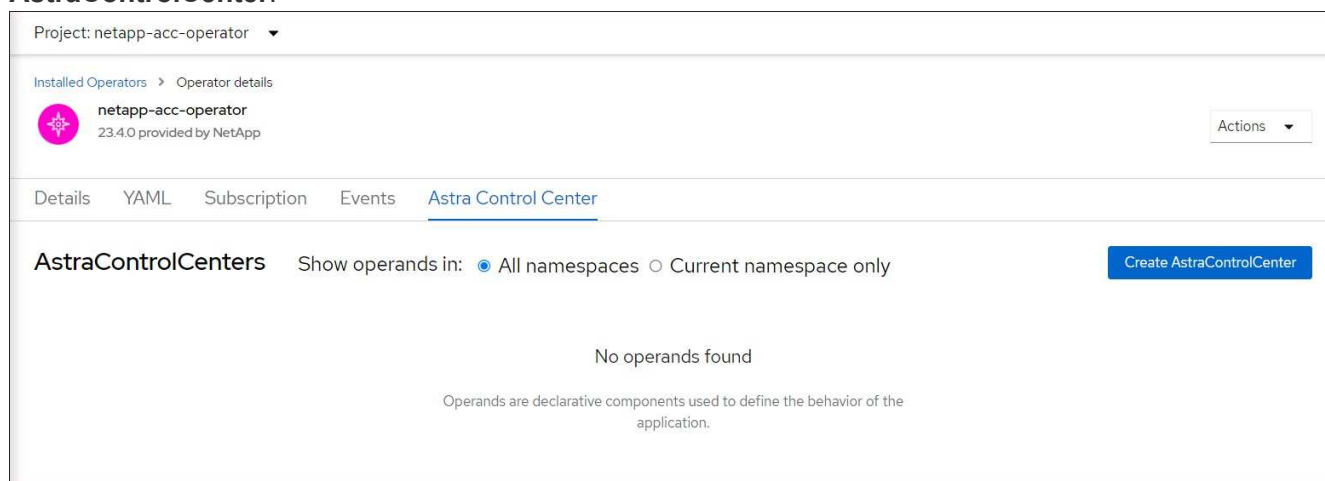


Se selecionou uma estratégia de aprovação manual, será-lhe pedido que aprove o plano de instalação manual para este operador.

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

Instale o Astra Control Center

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



- Preencha o `Create AstraControlCenter` campo do formulário:
 - Mantenha ou ajuste o nome do Astra Control Center.
 - Adicione etiquetas para o Astra Control Center.
 - Ative ou desative o suporte automático. Recomenda-se a manutenção da funcionalidade de suporte automático.
 - Insira o FQDN ou o endereço IP do Centro de Controle Astra. Não introduza `http://` ou `https://` no campo de endereço.
 - Digite a versão do Astra Control Center; por exemplo, 23.04.2-7.
 - Insira um nome de conta, endereço de e-mail e sobrenome do administrador.
 - Escolha uma política de recuperação de volume de `Retain`, `Recycle` ou `Delete`. O valor padrão é `Retain`.
 - Selecione o `scaleSize` 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**, insira seu caminho de Registro de imagem de contentor local. 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 registro 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 do Kubernetes no local ou em um dos clusters do 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 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 4,8



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 os seguintes recursos, além dos requisitos de recursos do ambiente:

Componente	Requisito
Capacidade de storage do NetApp Cloud Volumes ONTAP no back-end	Pelo menos 300GB disponível
Nós de trabalho (requisito AWS EC2)	No total, pelo menos 3 nós de trabalho, com 4 núcleos vCPU e 12GB GB de RAM cada
Balanceador de carga	Tipo de serviço "LoadBalancer" disponível para envio de tráfego de entrada para serviços no cluster do ambiente operacional
FQDN	Um método para apontar o FQDN do Astra Control Center para o endereço IP balanceado de carga
Astra Trident (instalado como parte da descoberta de clusters do Kubernetes no NetApp BlueXP , anteriormente chamado Gerenciador de nuvem)	Astra Trident 21,04 ou mais recente instalado e configurado e o NetApp ONTAP versão 9,5 ou mais recente como um back-end de storage

Componente	Requisito
Registro de imagens	Você precisa ter um Registro privado existente, como o AWS Elastic Container Registry, para o qual você pode enviar imagens de compilação do Astra Control Center. Você precisa fornecer o URL do Registro de imagens onde você vai carregar as imagens.  O cluster hospedado do Astra Control Center e o cluster gerenciado devem ter acesso ao mesmo Registro de imagem para poder fazer backup e restaurar aplicativos usando a imagem baseada em Restic.
Configuração Astra Trident/ONTAP	O Astra Control Center exige que uma classe de storage seja criada e definida como a classe de storage padrão. O Astra Control Center é compatível com as seguintes classes de storage do ONTAP Kubernetes criadas quando você importa o cluster do Kubernetes para o NetApp BlueXP (anteriormente conhecido como Cloud Manager). Eles são fornecidos pelo Astra Trident: • <code>vsaworkingenvironment-<>-ha-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-ha-san</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-san</code> <code>csi.trident.netapp.io</code>



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.



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 conector 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, criar um bucket do AWS S3, criar um ECR (Elastic Container Register) para hospedar as imagens do Astra Control Center 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. Crie um AWS Elastic Container Registry (ECR) para hospedar todas as imagens do ACC.



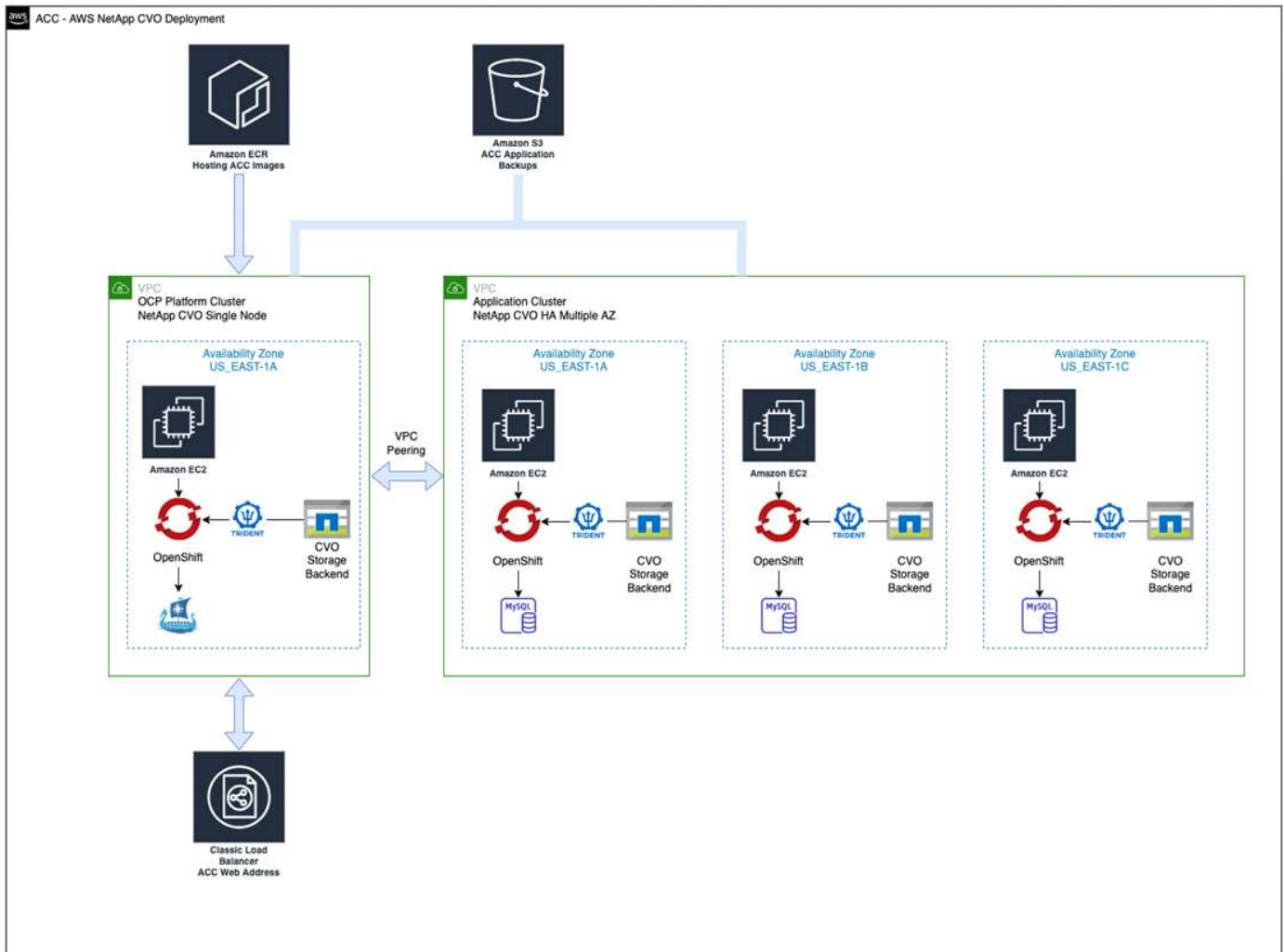
Se você não criar o ECR, o Astra Control Center não poderá acessar dados de monitoramento de um cluster que contém o Cloud Volumes ONTAP com um back-end da AWS. O problema é causado quando o cluster que você tenta descobrir e gerenciar usando o Astra Control Center não tem acesso ao AWS ECR.

6. Envie as imagens ACC 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 conector 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 conector na AWS usando o BlueXP"](#)

Passos

1. Adicione suas credenciais ao BlueXP.
2. Criar um espaço de trabalho.
3. Adicione um conector 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 conectará 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 Trident.
- 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 Trident 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 na 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) 4,10
- 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 os seguintes recursos, além dos requisitos de recursos do ambiente:

Componente	Requisito
Capacidade de storage do NetApp Cloud Volumes ONTAP no back-end	Pelo menos 300GB disponível

Componente	Requisito
Nós de trabalho (requisito de computação do GCP)	No total, pelo menos 3 nós de trabalho, com 4 núcleos vCPU e 12GB GB de RAM cada
Balanceador de carga	Tipo de serviço "LoadBalancer" disponível para envio de tráfego de entrada para serviços no cluster do ambiente operacional
FQDN (ZONA DNS DO GCP)	Um método para apontar o FQDN do Astra Control Center para o endereço IP balanceado de carga
Astra Trident (instalado como parte da descoberta de clusters do Kubernetes no NetApp BlueXP , anteriormente chamado Gerenciador de nuvem)	Astra Trident 21,04 ou mais recente instalado e configurado e o NetApp ONTAP versão 9,5 ou mais recente como um back-end de storage
Registo de imagens	Você deve ter um Registro privado existente, como o Google Container Registry, para o qual você pode enviar imagens de compilação do Astra Control Center. Você precisa fornecer o URL do Registro de imagens onde você vai carregar as imagens.  Você precisa habilitar o acesso anônimo para extrair imagens Restic para backups.
Configuração Astra Trident/ONTAP	O Astra Control Center exige que uma classe de storage seja criada e definida como a classe de storage padrão. O Astra Control Center é compatível com as seguintes classes de storage do ONTAP Kubernetes criadas quando você importa o cluster do Kubernetes para o NetApp BlueXP . Eles são fornecidos pelo Astra Trident: • <code>vsaworkingenvironment-<>-ha-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-ha-san</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-san</code> <code>csi.trident.netapp.io</code>



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.

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 conector 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, criar um Google Cloud Object Storage, criar um Google Container Register para hospedar as imagens do Astra Control Center 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. Crie um Registro de contêiner do Google para hospedar todas as imagens do Astra Control Center.
7. Configure o acesso do Google Container Registry para push/pull do Docker para todas as imagens do Astra Control Center.

Exemplo: As imagens ACC podem ser enviadas para este registo introduzindo 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=astra-control-center-<version>.manifest
GCP_CR_REGISTRY=<target image repository>
ASTRA_REGISTRY=<source ACC image repository>

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 < astra-control-center-22.04.41.manifest
```

8. Configurar zonas DNS.

Configurar o NetApp BlueXP para GCP

Usando o NetApp BlueXP , crie uma área de trabalho, adicione um conector 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 conector para o GCP.
 - a. Escolha "GCP" como Provedor.
 - b. Insira as credenciais do GCP. ["Criando um conector no GCP a partir do BlueXP"](#)Consulte .
 - c. Certifique-se de que o conector está a funcionar e mude para esse conector.
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 conectará 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 do Trident.
 - 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 Trident 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) 4,8
- 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 os seguintes recursos, além dos requisitos de recursos do ambiente:

["Requisitos do ambiente operacional do Astra Control Center"](#)Consulte a .

Componente	Requisito
Capacidade de storage do NetApp Cloud Volumes ONTAP no back-end	Pelo menos 300GB disponível
Nós de trabalho (requisito de computação do Azure)	No total, pelo menos 3 nós de trabalho, com 4 núcleos vCPU e 12GB GB de RAM cada
Balanceador de carga	Tipo de serviço "LoadBalancer" disponível para envio de tráfego de entrada para serviços no cluster do ambiente operacional
FQDN (zona DNS do Azure)	Um método para apontar o FQDN do Astra Control Center para o endereço IP balanceado de carga
Astra Trident (instalado como parte da descoberta de clusters do Kubernetes no NetApp BlueXP)	O Astra Trident 21,04 ou mais recente instalado e configurado e o NetApp ONTAP versão 9,5 ou mais recente serão usados como um back-end de storage
Registo de imagens	Você deve ter um Registro privado existente, como o Azure Container Registry (ACR), para o qual você pode enviar imagens de compilação do Astra Control Center. Você precisa fornecer o URL do Registro de imagens onde você vai carregar as imagens.  Você precisa habilitar o acesso anônimo para extrair imagens Restic para backups.

Componente	Requisito
Configuração Astra Trident/ONTAP	O Astra Control Center exige que uma classe de storage seja criada e definida como a classe de storage padrão. O Astra Control Center é compatível com as seguintes classes de storage do ONTAP Kubernetes criadas quando você importa o cluster do Kubernetes para o NetApp BlueXP . Eles são fornecidos pelo Astra Trident: • <code>vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-ha-san csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-nas csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-san csi.trident.netapp.io</code>



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.

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, criar um contentor Blob do Azure, criar um ACR (Registro de contentor do Azure) para hospedar as imagens do Astra Control Center 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. Crie um ACR (Azure Container Registry) para hospedar todas as imagens do Astra Control Center.
8. Configure o acesso ACR para o Docker push/pull de todas as imagens do Astra Control Center.
9. Empurre as imagens ACC para este registo introduzindo o seguinte script:

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

Exemplo:

```
manifestfile=astra-control-center-<version>.manifest
AZ_ACR_REGISTRY=<target image repository>
ASTRA_REGISTRY=<source ACC image repository>

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 < astra-control-center-22.04.41.manifest
```

10. 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 conector 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

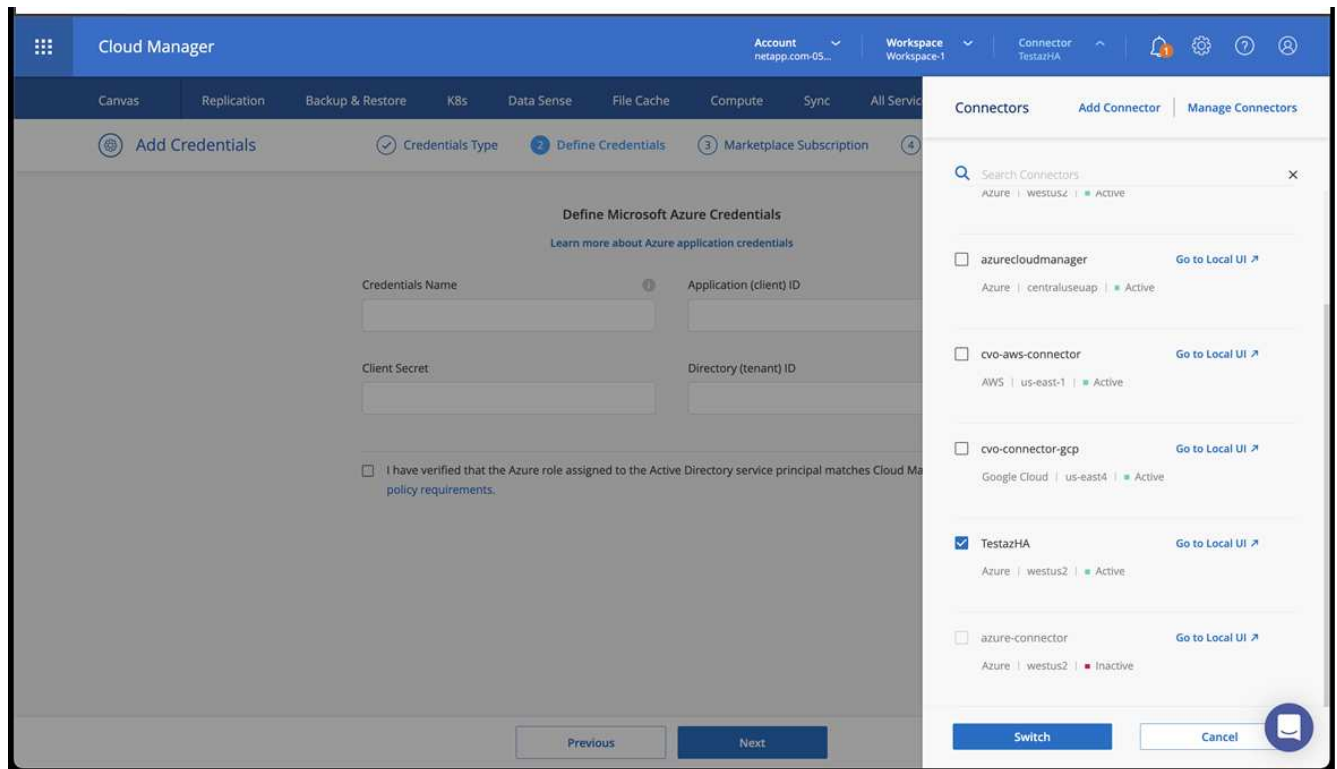
Acesso à conta do Azure com as permissões e funções necessárias do IAM

Passos

1. Adicione suas credenciais ao BlueXP .
2. Adicione um conector 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).

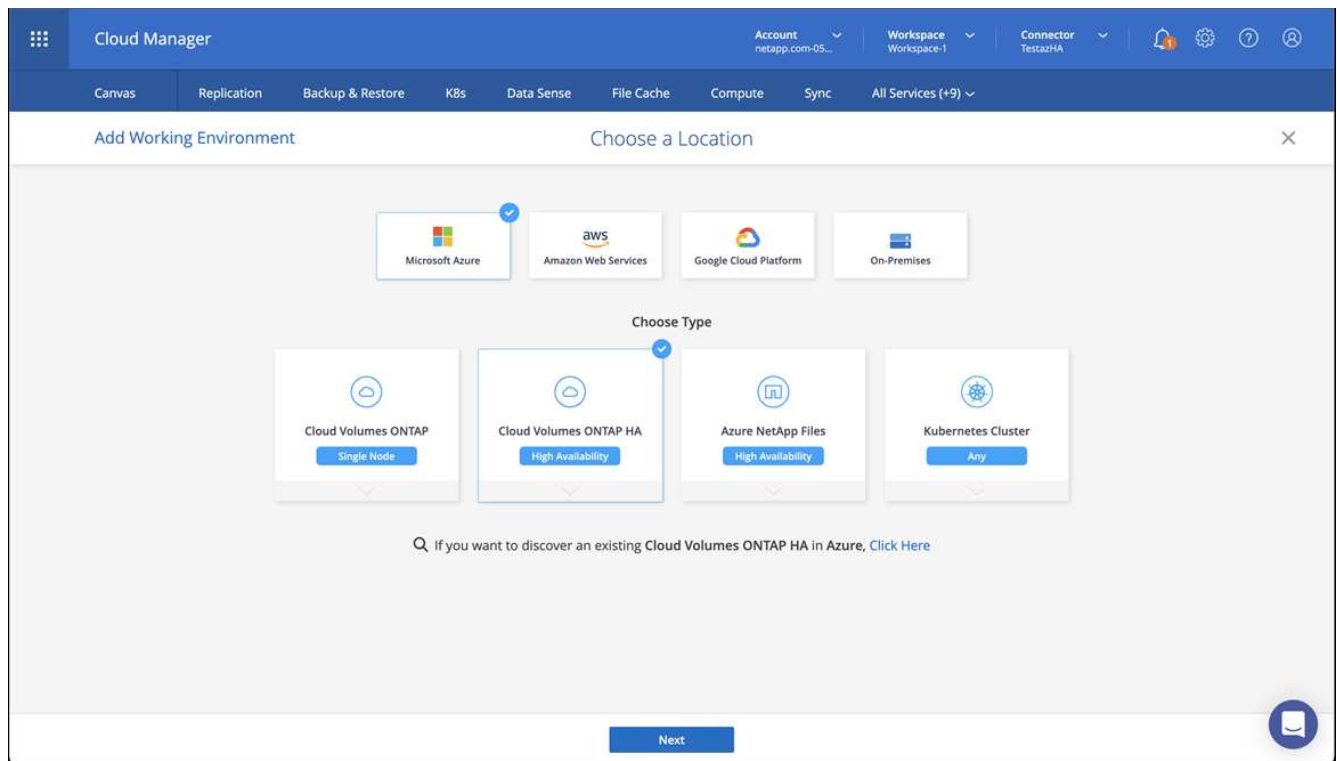
```
https://docs.netapp.com/us-
en/occm/task_creating_connectors_azure.html["Criando um conector no
Azure a partir do BlueXP"]Consulte .
```

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



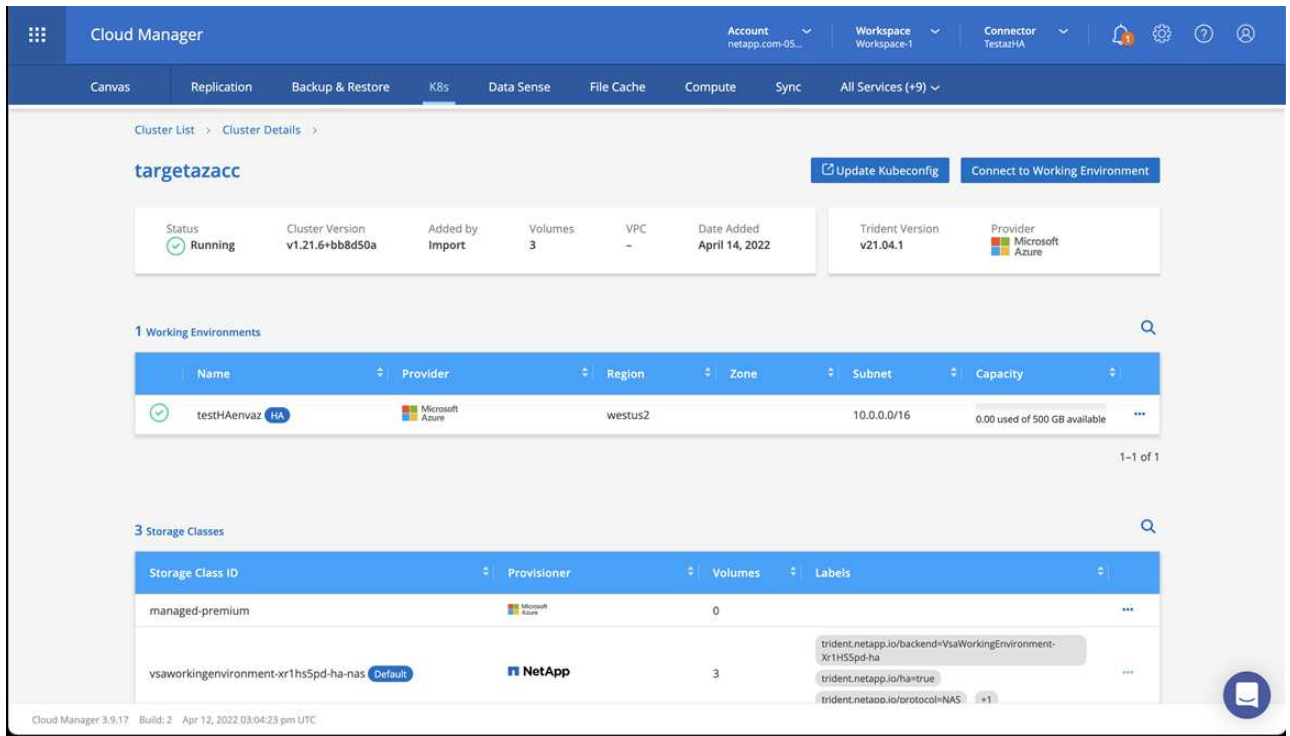
4. Crie um ambiente de trabalho para seu ambiente de nuvem.

- 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 Trident.

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 Trident 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]
```

Ative a comunicação de rede entre namespaces

Alguns ambientes usam construções NetworkPolicy para restringir o tráfego entre namespaces. O operador Astra Control Center e o Astra Control Center estão em diferentes namespaces. Os serviços nesses diferentes namespaces precisam ser capazes de se comunicar uns com os outros. Para ativar esta comunicação, siga estes passos.

Passos

1. Exclua todos os recursos da NetworkPolicy que existem no namespace Astra Control Center:

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

2. Para cada objeto NetworkPolicy retornado pelo comando anterior, use o seguinte comando para excluí-lo. Substitua [OBJECT_NAME] pelo nome do objeto retornado:

```
kubectl delete networkpolicy [OBJECT_NAME] -n [netapp-acc or custom namespace]
```

3. Aplique o seguinte arquivo de recurso para configurar o acc-avp-network-policy objeto para permitir que os serviços de plug-in Astra façam solicitações aos serviços do Astra Control Center. Substitua as informações entre parêntesis> por informações do seu ambiente:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: acc-avp-network-policy
  namespace: <ACC_NAMESPACE_NAME> # REPLACE THIS WITH THE ASTRA CONTROL
CENTER NAMESPACE NAME
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
    - from:
      - namespaceSelector:
          matchLabels:
            kubernetes.io/metadata.name: <PLUGIN_NAMESPACE_NAME> #
REPLACE THIS WITH THE ASTRA PLUGIN NAMESPACE NAME
```


4. Aplique o seguinte arquivo de recurso para configurar o `acc-operator-network-policy` objeto para permitir que o operador Astra Control Center se comunique com os serviços Astra Control Center. Substitua as informações entre parêntesis por informações do seu ambiente:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: acc-operator-network-policy
  namespace: <ACC_NAMESPACE_NAME> # REPLACE THIS WITH THE ASTRA CONTROL
CENTER NAMESPACE NAME
spec:
  podSelector: {}
  policyTypes:
    - Ingress
  ingress:
    - from:
      - namespaceSelector:
          matchLabels:
            kubernetes.io/metadata.name: <NETAPP-ACC-OPERATOR> #
REPLACE THIS WITH THE OPERATOR NAMESPACE NAME
```

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. Você pode 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>
....

#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>
....

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:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: <certificate-name>
  namespace: <ACC-deployment-namespace>
spec:
  secretName: <certificate-secret-name>
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  dnsNames:
  - <astra.dnsname.example.com> #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>
....

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 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.containo.us -n <ACC-deployment-namespace>
....

# tls:
#   options:
#     name: default
#     secretName: secure-testing-cert
#   store:
#     name: default

tls:
  options:
    name: default
  secretName: <certificate-secret-name>
  store:
    name: default
```

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

Informações sobre direitos autorais

Copyright © 2025 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTES DOCUMENTOS. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSALIENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTES DOCUMENTOS, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

Informações sobre marcas comerciais

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.