Instale o Astra Control Center

Astra Control Center

NetApp
August 11, 2025

This PDF was generated from https://docs.netapp.com/pt-br/astra-control-center-2310/get-started/cert-manager-prereqs.html on August 11, 2025. Always check docs.netapp.com for the latest.

Índice

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 conetados à Internet ou com conexão via rede.

Expanda para outros procedimentos de instalação

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

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

Antes de começar

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

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

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

```
kubectl get apiservices
```

- **Certifique-se de que um FQDN roteável**: O FQDN Astra que você planeja usar pode ser roteado para o cluster. Isso significa que você tem uma entrada DNS no seu servidor DNS interno ou está usando uma rota URL principal que já está registrada.
- **Configure cert Manager**: Se um gerenciador de cert já existir no cluster, você precisará executar alguns "etapas de pré-requisito" para que o Astra Control Center não tente instalar seu próprio gerenciador de cert. Por padrão, o Astra Control Center instala seu próprio gerenciador de cert durante a instalação.
- **Acesse o Registro de imagem do NetApp Astra Control**: Você tem a opção de obter imagens de instalação e melhorias de funcionalidade para o Astra Control, como o Astra Control Provisioner, a partir do Registro de imagens do NetApp.

Expanda para obter passos

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

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

b. Na mesma página, selecione **Generate API token** e copie a cadeia de token da API para a área de transferência e salve-a no seu editor.

c. Faça login no Registro do Astra Control:

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

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

Detalhes de malha de serviço do Istio

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

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

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

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

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

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

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

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

Faça download e extraia Astra Control Center

Você pode optar por baixar o pacote Astra Control Center do site de suporte da NetApp ou usar o Docker para extrair o pacote do Registro de imagem do serviço Astra Control.

Site de suporte da NetApp

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

   Expanda para obter detalhes

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

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

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

3. Extraia as imagens do pacote Astra Control Center:

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

Registro de imagem Astra Control

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

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:

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

```
ls kubectl-astra/
```

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

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

Adicione as imagens ao seu registo 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 o `acc.manifest.bundle.yaml` arquivo e estes diretórios:

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

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

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

Podman

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

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

2. Inicie sessão no seu registo:

   ```
   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=23.10.0-68
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/${
PACKAGEVERSION}/${astraImageNoPath}
done
```

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

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.10.0-68
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/${
PACKAGEVERSION}/${astraImageNoPath}
done
```

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

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/23.10.0-68/image:version
```

Configure namespace e segredo para Registros com requisitos de autenticação

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

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

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

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

Expanda para obter passos

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

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

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

> O marcador de posição `your_registry_path` deve corresponder à localização das imagens que carregou anteriormente (por exemplo, `[Registry_URL]/netapp/astra/astracc/23.10.0-68`).

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

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

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

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

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]
```

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

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

Expanda para amostra astra_control_center_operator_deploy.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: ASTRA_IMAGE_REGISTRY/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_INSTALLTIMEOUT
          value: 5m
        image: ASTRA_IMAGE_REGISTRY/acc-operator:23.10.72
        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
```

Expandir para resposta da amostra:

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

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 carateres para o nome que deseja associar à conta Astra Control Center. Só pode haver uma accountName.	cadeia de carateres	`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 carateres	`23.10.0-68`

<code> </code>

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

<code> AutoSupport </code>

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

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

<code> email</code>

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

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

<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 `kubectl get sc` para determinar suas classes de armazenamento configuradas existentes. Uma das classes de storage baseadas no Astra Trident deve ser inserida no arquivo MANIFEST (`astra-control-center-<version>.manifest`) e será usada para PVS Astra. Se não estiver definida, a classe de armazenamento padrão será usada. Nota: Se uma classe de armazenamento padrão estiver configurada, certifique-se de que é a única classe de armazenamento que tem a anotação padrão.	cadeia de carateres	`ontap-gold`

<code> volume ReclaimPolicy</code>

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

<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. IMPORTANTE: Se você pretende usar uma malha de serviço com o Astra Control Center, você deve `Generic` selecionar como tipo de ingresso e configurar o seu próprio "controlador de entrada". **`AccTraefik(ingressType: "AccTraefik"`**) Utilize esta opção quando preferir não configurar um controlador de entrada. Isso implanta o gateway Astra Control Center `traefik` como um serviço do tipo Kubernetes LoadBalancer. O Astra Control Center usa um serviço do tipo "LoadBalancer" (`svc/traefik` no namespace Astra Control Center) e exige que seja atribuído um endereço IP externo acessível. Se os balanceadores de carga forem permitidos em seu ambiente e você ainda não tiver um configurado, você poderá usar o MetalLB	cadeia de carateres	• `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 carateres	• `Small` • `Medium` (Este é o valor padrão)

<code>astraResourcesScaler</code>

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

<code> AdditionalValues</code>

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

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

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

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

Expanda para amostra astra_control_center.yaml

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

Instalação completa do operador e do Centro de Controle Astra

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

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

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

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

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

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

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

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

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

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

Verifique o status do sistema

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

Passos

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

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

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

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

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

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

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

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

Expandir para resposta de amostra

```
NAME                                          READY   STATUS
RESTARTS     AGE
acc-helm-repo-6cc7696d8f-pmhm8                1/1     Running     0
9h
activity-597fb656dc-5rd4l                     1/1     Running     0
9h
activity-597fb656dc-mqmcw                     1/1     Running     0
9h
api-token-authentication-62f84                1/1     Running     0
9h
api-token-authentication-68nlf                1/1     Running     0
9h
api-token-authentication-ztgrm                1/1     Running     0
9h
asup-669d4ddbc4-fnmwp                         1/1     Running     1
(9h ago)   9h
authentication-78789d7549-lk686               1/1     Running     0
9h
bucketservice-65c7d95496-24x7l                1/1     Running     3
(9h ago)   9h
cert-manager-c9f9fbf9f-k8zq2                  1/1     Running     0
9h
cert-manager-c9f9fbf9f-qjlzm                  1/1     Running     0
9h
cert-manager-cainjector-dbbbd8447-b5qll       1/1     Running     0
9h
cert-manager-cainjector-dbbbd8447-p5whs       1/1     Running     0
9h
cert-manager-webhook-6f97bb7d84-4722b         1/1     Running     0
9h
cert-manager-webhook-6f97bb7d84-86kv5         1/1     Running     0
9h
certificates-59d9f6f4bd-2j899                 1/1     Running     0
9h
certificates-59d9f6f4bd-9d9k6                 1/1     Running     0
9h
certificates-expiry-check-28011180--1-8lkxz   0/1     Completed   0
9h
cloud-extension-5c9c9958f8-jdhrp              1/1     Running     0
9h
cloud-insights-service-5cdd5f7f-pp8r5         1/1     Running     0
9h
composite-compute-66585789f4-hxn5w            1/1     Running     0
9h
```

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

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

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

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

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

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

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

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

Resposta:

```
NAME    UUID                                   VERSION     ADDRESS
READY
astra   9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f  23.10.0-68
10.111.111.111  True
```

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

Configure a entrada para o balanceamento de carga

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

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

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

Antes de começar

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

Etapas para a entrada do Istio

1. Configurar a entrada do Istio.

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

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

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

   Exemplo de comando:

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

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

   Exemplo de comando:

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

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

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

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

5. Aplicar as alterações:

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

6. Verifique o estado da entrada:

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

Resposta:

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

7. Concluir a instalação do Astra Control Center.

Etapas para o controlador nginx Ingress

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

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

3. Aplicar as alterações:

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

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

Passos para o controlador OpenShift Ingress

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

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

Faça login na IU do Astra Control Center

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

Passos

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

   Você pode criar um certificado personalizado após o login.

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

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

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

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

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

Solucionar problemas da instalação

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

Opções

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

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

- Para verificar a saída do Astra Control Center CR:

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

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.

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 DESTE DOCUMENTO. 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 SOBRESSALENTES; 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 DESTE SOFTWARE, 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.