



Présentation de l'installation

Astra Control Center

NetApp
November 21, 2023

Sommaire

- Présentation de l'installation 1
 - Installer le centre de contrôle Astra en suivant la procédure standard 1
 - Installez Astra Control Center à l'aide d'OpenShift OperatorHub 38
 - Installer le centre de contrôle Astra avec un système de stockage back-end Cloud Volumes ONTAP 46
 - Configurer le centre de contrôle Astra après l'installation 61

Présentation de l'installation

Choisissez l'une des procédures d'installation suivantes du centre de contrôle Astra :

- ["Installer le centre de contrôle Astra en suivant la procédure standard"](#)
- ["\(Si vous utilisez Red Hat OpenShift\) installez Astra Control Center à l'aide d'OpenShift OperatorHub"](#)
- ["Installer le centre de contrôle Astra avec un système de stockage back-end Cloud Volumes ONTAP"](#)

Selon votre environnement, une configuration supplémentaire peut être nécessaire après l'installation d'Astra Control Center :

- ["Configurer le centre de contrôle Astra après l'installation"](#)

Installer le centre de contrôle Astra en suivant la procédure standard

Pour installer Astra Control Center, téléchargez le bundle d'installation depuis le site de support NetApp et effectuez les opérations suivantes. Vous pouvez utiliser cette procédure pour installer Astra Control Center dans des environnements connectés à Internet ou équipés d'un filtre à air.

Autres procédures d'installation

- **Installer avec RedHat OpenShift OperatorHub:** Utilisez ceci ["autre procédure"](#) Pour installer Astra Control Center sur OpenShift à l'aide d'OperatorHub.
- **Installer dans le Cloud public avec Cloud Volumes ONTAP backend:** Utilisez ["ces procédures"](#) Pour installer Astra Control Center dans Amazon Web Services (AWS), Google Cloud Platform (GCP) ou Microsoft Azure avec un système de stockage principal Cloud Volumes ONTAP.

Pour une démonstration du processus d'installation d'Astra Control Center, reportez-vous à la section ["vidéo"](#).

Avant de commencer

- ["Avant de commencer l'installation, préparez votre environnement pour le déploiement d'Astra Control Center"](#).
- Si vous avez configuré ou que vous souhaitez configurer des stratégies de sécurité de pod dans votre environnement, familiarisez-vous avec les stratégies de sécurité de pod et leur incidence sur l'installation d'Astra Control Center. Reportez-vous à la section ["Comprendre les restrictions de la stratégie de sécurité du pod"](#).
- Assurez-vous que tous les services API sont en état de santé et disponibles :

```
kubectl get apiservices
```

- Assurez-vous que le FQDN Astra que vous prévoyez d'utiliser est routable vers ce cluster. Cela signifie que vous avez une entrée DNS dans votre serveur DNS interne ou que vous utilisez une route URL de base déjà enregistrée.
- Si un cert Manager existe déjà dans le cluster, vous devez en effectuer certaines ["étapes préalables"](#) Pour

qu'Astra Control Center ne tente pas d'installer son propre gestionnaire de certificat. Par défaut, Astra Control Center installe son propre gestionnaire de certificats lors de l'installation.



Déployez Astra Control Center dans un troisième domaine de panne ou sur un site secondaire. Cela est recommandé pour la réplication d'applications et la reprise sur incident transparente.

Description de la tâche

Le processus d'installation d'Astra Control Center vous aide à :

- Poser les composants Astra dans le `netapp-acc` (ou espace de nom personnalisé).
- Créez un compte d'administrateur propriétaire Astra Control par défaut.
- Définissez une adresse e-mail d'utilisateur administratif et un mot de passe de configuration initiale par défaut. Ce rôle de propriétaire est attribué à cet utilisateur pour la première connexion à l'interface utilisateur.
- Vérifiez que toutes les POD Astra Control Center sont en cours d'exécution.
- Installez l'interface utilisateur du centre de contrôle Astra.



Ne supprimez pas l'opérateur du centre de contrôle Astra (par exemple, `kubectl delete -f astra_control_center_operator_deploy.yaml`) À tout moment pendant l'installation ou le fonctionnement d'Astra Control Center pour éviter de supprimer les modules.

Étapes

Pour installer le centre de contrôle Astra, procédez comme suit :

- [Téléchargez et extrayez Astra Control Center](#)
- [Installez le plug-in NetApp Astra kubectl](#)
- [Ajoutez les images à votre registre local](#)
- [Configurez l'espace de noms et le secret pour les registres avec les exigences d'authentification](#)
- [Poser le conducteur du centre de commande Astra](#)
- [Configurer le centre de contrôle Astra](#)
- [Installation complète du centre de contrôle Astra et du conducteur](#)
- [Vérifiez l'état du système](#)
- [Configurer l'entrée pour l'équilibrage de charge](#)
- [Connectez-vous à l'interface utilisateur du centre de contrôle Astra](#)

Téléchargez et extrayez Astra Control Center

1. Accédez au "[Page de téléchargements d'Astra Control Center](#)" Sur le site de support NetApp.
2. Téléchargez le pack contenant Astra Control Center (`astra-control-center-[version].tar.gz`).
3. (Recommandé mais facultatif) Téléchargez le lot de certificats et de signatures pour Astra Control Center (`astra-control-center-certs-[version].tar.gz`) pour vérifier la signature du paquet :

```
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
```

La sortie s'affiche `Verified OK` une fois la vérification terminée.

4. Extraire les images du pack Astra Control Center :

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

Installez le plug-in NetApp Astra kubectl

Vous pouvez utiliser le plug-in de ligne de commande NetApp Astra kubectl pour envoyer les images vers un référentiel Docker local.

Avant de commencer

NetApp fournit des binaires de plug-ins pour différentes architectures CPU et systèmes d'exploitation. Avant d'effectuer cette tâche, vous devez savoir quelle unité centrale et quel système d'exploitation vous possédez.

Si vous avez déjà installé le plug-in à partir d'une installation précédente, "[vérifiez que vous disposez de la dernière version](#)" avant d'effectuer ces étapes.

Étapes

1. Répertoriez les binaires NetApp Astra kubectl disponibles et notez le nom du fichier dont vous avez besoin pour votre système d'exploitation et votre architecture de processeur :



La bibliothèque de plug-ins kubectl fait partie du bundle tar et est extraite dans le dossier `kubectl-astra`.

```
ls kubectl-astra/
```

2. Déplacez le bon binaire dans le chemin actuel et renommez-le `kubectl-astra`:

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

Ajoutez les images à votre registre local

1. Suivez la séquence d'étapes appropriée pour votre moteur de mise en conteneurs :

Docker

1. Accédez au répertoire racine du tarball. Vous devriez voir ce fichier et ce répertoire:

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

2. Envoyez les images du package dans le répertoire d'images Astra Control Center vers votre registre local. Effectuez les remplacements suivants avant d'exécuter le `push-images` commande :

- Remplacez `<BUNDLE_FILE>` par le nom du fichier bundle Astra Control (`acc.manifest.bundle.yaml`).
- Remplacer `<MY_FULL_REGISTRY_PATH>` par l'URL du référentiel Docker, par exemple "`<a href="https://<docker-registry>" class="bare">https://<docker-registry>"`".
- Remplacez `<MY_REGISTRY_USER>` par le nom d'utilisateur.
- Remplacez `<MY_REGISTRY_TOKEN>` par un jeton autorisé pour le registre.

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

Podman

1. Accédez au répertoire racine du tarball. Vous devriez voir ce fichier et ce répertoire:

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

2. Connectez-vous à votre registre :

```
podman login <YOUR_REGISTRY>
```

3. Préparez et exécutez l'un des scripts suivants qui est personnalisé pour la version de Podman que vous utilisez. Remplacez `<MY_FULL_REGISTRY_PATH>` par l'URL de votre référentiel qui inclut tous les sous-répertoires.

```
<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

```



Le chemin d'accès à l'image que le script crée doit ressembler aux éléments suivants, selon la configuration de votre registre :

```

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

```

Configurez l'espace de noms et le secret pour les registres avec les exigences d'authentification

1. Exporter le KUBECONFIG pour le groupe hôte du centre de contrôle Astra :

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



Avant de terminer l'installation, assurez-vous que votre KUBECONFIG pointe vers le groupe d'instruments où vous souhaitez installer le centre de contrôle Astra. Le KUBECONFIG ne peut contenir qu'un seul contexte.

2. Si vous utilisez un registre qui nécessite une authentification, vous devez procéder comme suit :

a. Créer le `netapp-acc-operator` espace de noms :

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

Réponse :

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

b. Créez un secret pour le `netapp-acc-operator` espace de noms. Ajoutez des informations sur Docker et exécutez la commande suivante :



Le paramètre fictif `your_registry_path` doit correspondre à l'emplacement des images que vous avez téléchargées précédemment (par exemple, `[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]
```

Exemple de réponse :

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



Si vous supprimez l'espace de noms après la génération du secret, recréez l'espace de noms, puis régénérez le secret pour l'espace de noms.

c. Créer le `netapp-acc` (ou espace de nom personnalisé).

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

Exemple de réponse :


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

- d. Créez un secret pour le netapp-acc (ou espace de nom personnalisé). Ajoutez des informations sur Docker et exécutez la commande suivante :

```
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]
```

Réponse

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

Poser le conducteur du centre de commande Astra

1. Modifier le répertoire :

```
cd manifests
```

2. Modifiez le YAML de déploiement de l'opérateur Astra Control Center (`astra_control_center_operator_deploy.yaml`) pour faire référence à votre registre local et à votre secret.

```
vim astra_control_center_operator_deploy.yaml
```



Un échantillon annoté YAML suit ces étapes.

- a. Si vous utilisez un registre qui nécessite une authentification, remplacez la ligne par défaut de `imagePullSecrets: []` avec les éléments suivants :

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

- b. Changez `[your_registry_path]` pour le `kube-rbac-proxy` image dans le chemin du registre où vous avez poussé les images dans un [étape précédente](#).
- c. Changez `[your_registry_path]` pour le `acc-operator-controller-manager` image dans le chemin du registre où vous avez poussé les images dans un [étape précédente](#).

```
<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. Poser le conducteur du centre de commande Astra :

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

Exemple de réponse :

```
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. Vérifiez que les pods sont en cours d'exécution :

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

Configurer le centre de contrôle Astra

1. Modifiez le fichier de ressources personnalisées (CR) Astra Control Center (`astra_control_center.yaml`) pour créer des comptes, un support, un registre et d'autres configurations nécessaires :

```
vim astra_control_center.yaml
```



Un échantillon annoté YAML suit ces étapes.

2. Modifiez ou confirmez les paramètres suivants :

`<code>accountName</code>`

Réglage	Guidage	Type	Exemple
accountName	Modifiez le <code>accountName</code> Chaîne du nom que vous souhaitez associer au compte Astra Control Center. Il ne peut y avoir qu'un seul nom de compte.	chaîne	Exemple

`<code>astraVersion</code>`

Réglage	Guidage	Type	Exemple
astraVersion	La version d'Astra Control Center à déployer. Aucune action n'est nécessaire pour ce paramètre car la valeur sera pré-remplie.	chaîne	23.04.2-7

`<code>astraAddress</code>`

Réglage	Guidage	Type	Exemple
astraAddress	<p>Modifiez le <code>astraAddress</code> Chaîne sur le FQDN (recommandé) ou l'adresse IP que vous souhaitez utiliser dans votre navigateur pour accéder à Astra Control Center. Cette adresse définit la façon dont Astra Control Center se trouve dans votre centre de données et est le même FQDN ou l'adresse IP que vous avez fournie à partir de votre équilibreur de charge une fois que vous avez terminé "Exigences du centre de contrôle Astra".</p> <p>REMARQUE : ne pas utiliser <code>http://</code> ou <code>https://</code> dans l'adresse. Copier ce FQDN pour l'utiliser dans un plus tard.</p>	chaîne	astra.example.com

<code>autoSupport</code>

Vos sélections dans cette section déterminent si vous allez participer à l'application de support proactif de NetApp, à NetApp Active IQ et à l'endroit où les données seront envoyées. Une connexion Internet est requise (port 442) et toutes les données de support sont anonymisées.

Réglage	Utiliser	Guidage	Type	Exemple
<code>autoSupport.enrolled</code>	Soit <code>enrolled</code> ou <code>url</code> les champs doivent être sélectionnés	Changer <code>enrolled</code> Pour <code>AutoSupport</code> à <code>false</code> pour les sites sans connexion internet ou sans conservation <code>true</code> pour les sites connectés. Un réglage de <code>true</code> Permet d'envoyer des données anonymes à NetApp à des fins d'assistance. La sélection par défaut est <code>false</code> Aucune donnée de support n'est envoyée à NetApp.	Booléen	<code>false</code> (cette valeur est la valeur par défaut)
<code>autoSupport.url</code>	Soit <code>enrolled</code> ou <code>url</code> les champs doivent être sélectionnés	Cette URL détermine l'emplacement d'envoi des données anonymes.	chaîne	https://support.netapp.com/asupprod/post/1.0/postAsup

<code>email</code>

Réglage	Guidage	Type	Exemple
email	Modifiez le email chaîne à l'adresse d'administrateur initiale par défaut. Copiez cette adresse e-mail pour l'utiliser dans un plus tard . Cette adresse e-mail sera utilisée comme nom d'utilisateur du compte initial pour se connecter à l'interface utilisateur et sera informée des événements dans Astra Control.	chaîne	admin@example.com

<code>firstName</code>

Réglage	Guidage	Type	Exemple
firstName	Prénom de l'administrateur initial par défaut associé au compte Astra. Le nom utilisé ici sera visible dans un en-tête de l'interface utilisateur après votre première connexion.	chaîne	SRE

<code>LastName</code>

Réglage	Guidage	Type	Exemple
lastName	Nom de l'administrateur initial par défaut associé au compte Astra. Le nom utilisé ici sera visible dans un en-tête de l'interface utilisateur après votre première connexion.	chaîne	Admin

<code>imageRegistry</code>

Vos sélections dans cette section définissent le registre d'images du conteneur qui héberge les images d'application Astra, l'opérateur du centre de contrôle Astra et le référentiel Helm d'Astra Control Center.

Réglage	Utiliser	Guidage	Type	Exemple
<code>imageRegistry.name</code>	Obligatoire	Nom du registre d'images dans lequel vous avez poussé les images dans le étape précédente . Ne pas utiliser <code>http://</code> ou <code>https://</code> dans le nom du registre.	chaîne	<code>example.registry.com/astra</code>
<code>imageRegistry.secret</code>	Obligatoire si la chaîne que vous avez entrée pour <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> ligne comprise entre <code>imageRegistry</code> sinon, l'installation échouera.	Nom du secret Kubernetes utilisé pour s'authentifier auprès du registre d'images.	chaîne	<code>astra-registry-cred</code>

`<code>storageClass</code>`

Réglage	Guidage	Type	Exemple
<code>storageClass</code>	<p>Modifiez le <code>storageClass</code> valeur à partir de <code>ontap-gold</code> À une autre ressource de classe de stockage Astra Trident, comme requis par votre installation. Lancer la commande <code>kubectl get sc</code> pour déterminer vos classes de stockage configurées existantes. L'une des classes de stockage basées sur Astra Trident doit être saisie dans le fichier manifeste (<code>astra-control-center-<version>.manifest</code>) Et sera utilisé pour ASTRA PVS. Si elle n'est pas définie, la classe de stockage par défaut sera utilisée.</p> <p>REMARQUE : si une classe de stockage par défaut est configurée, assurez-vous qu'elle est la seule classe de stockage à avoir l'annotation par défaut.</p>	chaîne	<code>ontap-gold</code>

`<code>volumeReclaimPolicy</code>`

Réglage	Guidage	Type	Options
<code>volumeReclaimPolicy</code>	Cette règle définit la règle de récupération pour les volumes persistants d'Astra. Définition de cette règle sur <code>Retain</code> Conserve les volumes persistants après la suppression d'Astra. Définition de cette règle sur <code>Delete</code> supprime les volumes persistants après la suppression d'astra. Si cette valeur n'est pas définie, les PV sont conservés.	chaîne	<ul style="list-style-type: none">• <code>Retain</code> (Il s'agit de la valeur par défaut)• <code>Delete</code>

`<code>ingressType</code>`





Réglage	Guidage	Type	Options
ingressType	<p>Utilisez l'un des types d'entrées suivants :</p> <p>Generic (ingressType: "Generic") (Par défaut) Utilisez cette option si vous avez un autre contrôleur d'entrée en service ou si vous préférez utiliser votre propre contrôleur d'entrée. Après le déploiement du centre de contrôle Astra, vous devez configurer le "contrôleur d'entrée" Pour exposer Astra Control Center avec une URL.</p> <p>AccTraefik (ingressType: "AccTraefik") Utilisez cette option lorsque vous préférez ne pas configurer de contrôleur d'entrée. Ceci déploie le centre de contrôle Astra traefik Passerelle en tant que service de type Kubernetes LoadBalancer.</p> <p>Le centre de contrôle Astra utilise un service de type « équilibreur de charge » (svc/traefik Dans l'espace de noms du centre de contrôle Astra), et exige qu'il se voit attribuer une adresse IP externe accessible. Si des équilibreurs de charge sont autorisés dans votre environnement et que vous n'en avez pas encore configuré, vous pouvez utiliser MetalLB ou un autre équilibreur</p>	chaîne	<ul style="list-style-type: none"> • Generic (il s'agit de la valeur par défaut) • AccTraefik

`<code>scaleSize</code>`

Réglage	Guidage	Type	Options
<code>scaleSize</code>	<p>Par défaut, Astra utilisera la haute disponibilité (HA) <code>scaleSize</code> de <code>Medium</code>, Qui déploie la plupart des services en haute disponibilité et déploie plusieurs répliques pour assurer la redondance. Avec <code>scaleSize</code> comme <code>Small</code>, Astra réduira le nombre de répliques pour tous les services, à l'exception des services essentiels, afin de réduire la consommation.</p> <p>CONSEIL : <code>Medium</code> les déploiements se composent d'environ 100 pods (à l'exclusion des workloads transitoires). 100 modules sont basés sur une configuration à trois nœuds maîtres et trois nœuds workers). Tenez compte des contraintes de limite réseau par pod qui peuvent représenter un problème dans votre environnement, en particulier lors de l'examen des scénarios de reprise d'activité.</p>	chaîne	<ul style="list-style-type: none">• <code>Small</code>• <code>Medium</code> (Il s'agit de la valeur par défaut)

`<code>astraResourcesScaler</code>`

Réglage	Guidage	Type	Options
<code>astraResourcesScaler</code>	<p>Options d'évolutivité pour les limites de ressources AstrakControlCenter. Par défaut, Astra Control Center se déploie avec des demandes de ressources définies pour la plupart des composants d'Astra. Avec cette configuration, la pile logicielle Astra Control Center est plus performante dans les environnements soumis à une charge et à une évolutivité accrues des applications.</p> <p>Cependant, dans les scénarios utilisant des grappes de développement ou de test plus petites, le champ CR <code>astraResourcesScaler</code> peut être réglé sur <code>Off</code>. Cela désactive les demandes de ressources et permet un déploiement sur les clusters plus petits.</p>	chaîne	<ul style="list-style-type: none">• Default (Il s'agit de la valeur par défaut)• Off

<code>additionalValues</code>

- Pour les communications Astral Control Center et Cloud Insights, la vérification du certificat TLS est désactivée par défaut. Vous pouvez activer la vérification de certification TLS pour la communication entre Cloud Insights et le cluster hôte Astra Control Center et le cluster géré en ajoutant la section suivante à la `additionalValues`.

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

`<code>crds</code>`

Vos sélections dans cette section déterminent comment Astra Control Center doit traiter les CRD.

Réglage	Guidage	Type	Exemple
<code>crds.externalCertManager</code>	<p>Si vous utilisez un gestionnaire de certificats externe, modifiez-le <code>externalCertManager</code> à <code>true</code>. La valeur par défaut <code>false</code> Provoque l'installation d'Astra Control Center de ses propres CRD de <code>cert Manager</code> lors de l'installation.</p> <p>Les CRDS sont des objets à l'échelle du cluster et leur installation peut avoir un impact sur d'autres parties du cluster. Vous pouvez utiliser cet indicateur pour signaler à Astra Control Center que ces CRD seront installés et gérés par l'administrateur de cluster en dehors du centre de contrôle Astra.</p>	Booléen	<code>False</code> (cette valeur est la valeur par défaut)
<code>crds.externalTraefik</code>	<p>Par défaut, Astra Control Center installe les CRD Traefik requis. Les CRDS sont des objets à l'échelle du cluster et leur installation peut avoir un impact sur d'autres parties du cluster. Vous pouvez utiliser cet indicateur pour signaler à Astra Control Center que ces CRD seront installés et gérés par l'administrateur de cluster en dehors du centre de contrôle Astra.</p>	Booléen	<code>False</code> (cette valeur est la valeur par défaut)



Assurez-vous d'avoir sélectionné la classe de stockage et le type d'entrée appropriés pour votre configuration avant de terminer l'installation.

```
<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
```

Installation complète du centre de contrôle Astra et du conducteur

1. Si vous ne l'avez pas déjà fait dans une étape précédente, créez le `netapp-acc` (ou personnalisée) espace de noms :

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

Exemple de réponse :

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

2. Poser le centre de contrôle Astra dans le `netapp-acc` (ou votre espace de noms personnalisé) :

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

Exemple de réponse :

```
astracenter.astra.netapp.io/astra created
```



L'opérateur d'Astra Control Center effectue une vérification automatique des exigences de l'environnement. Manquant "[de formation](#)" Peut entraîner une défaillance de votre installation ou un dysfonctionnement d'Astra Control Center. Voir la [section suivante](#) pour vérifier la présence de messages d'avertissement liés au contrôle automatique du système.

Vérifiez l'état du système

Vous pouvez vérifier l'état du système à l'aide des commandes `kubectl`. Si vous préférez utiliser OpenShift, vous pouvez utiliser des commandes `oc` comparables pour les étapes de vérification.

Étapes

1. Vérifiez que le processus d'installation n'a pas produit de messages d'avertissement relatifs aux vérifications de validation :

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



Des messages d'avertissement supplémentaires sont également signalés dans les journaux de l'opérateur d'Astra Control Center.

2. Corrigez tous les problèmes de votre environnement qui ont été signalés par les vérifications automatisées des exigences.



Vous pouvez corriger les problèmes en vous assurant que votre environnement respecte les "[de formation](#)" Pour Astra Control Center.

3. Vérifiez que tous les composants du système sont correctement installés.

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

Chaque pod doit avoir un statut de `Running`. Le déploiement des modules du système peut prendre plusieurs minutes.

Exemple de réponse

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			
bucket-service-65c7d95496-24x7l	1/1	Running	3
(9h ago) 9h			
cert-manager-c9f9fbf9f-k8zq2	1/1	Running	0
9h			
cert-manager-c9f9fbf9f-qj1zm	1/1	Running	0
9h			
cert-manager-cainjector-dbbbd8447-b5q1l	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--1-81kxz	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-cxmxw	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 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-mk917 (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. (Facultatif) pour vous assurer que l'installation est terminée, vous pouvez regarder le `acc-operator` journaux utilisant la commande suivante.

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



`accHost` l'enregistrement du cluster est l'une des dernières opérations. en cas de défaillance, le déploiement ne pourra pas échouer. Dans l'éventualité où un échec d'enregistrement du cluster était indiqué dans les journaux, vous pouvez essayer de nouveau l'enregistrement via le ["Ajout du flux de travail du cluster dans l'interface utilisateur"](#) Ou API.

5. Lorsque tous les modules sont en cours d'exécution, vérifiez que l'installation a réussi (`READY` est `True`) Et obtenez le mot de passe de configuration initial que vous utiliserez lorsque vous vous connectez à Astra Control Center :

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

Réponse :

NAME	UUID	VERSION	ADDRESS
READY			
astra	9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f	23.04.2-7	10.111.111.111
True			



Copiez la valeur UUID. Le mot de passe est `ACC-` Suivi de la valeur UUID (`ACC-[UUID]` ou, dans cet exemple, `ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f`).

Configurer l'entrée pour l'équilibrage de charge

Vous pouvez configurer un contrôleur d'entrée Kubernetes qui gère l'accès externe aux services. Ces procédures fournissent des exemples de configuration pour un contrôleur d'entrée si vous avez utilisé la valeur par défaut de `ingressType: "Generic"` Dans la ressource personnalisée Astra Control Center (`astra_control_center.yaml`). Vous n'avez pas besoin d'utiliser cette procédure si vous avez spécifié `ingressType: "AccTraefik"` Dans la ressource personnalisée Astra Control Center (`astra_control_center.yaml`).

Après le déploiement du centre de contrôle Astra, vous devrez configurer le contrôleur d'entrée pour exposer le centre de contrôle Astra à une URL.

Les étapes de configuration varient en fonction du type de contrôleur d'entrée utilisé. Le centre de contrôle Astra prend en charge de nombreux types de contrôleurs d'entrée. Ces procédures de configuration fournissent des exemples pour les types de contrôleurs d'entrée suivants :

- Entrée Istio

- Contrôleur d'entrée Nginx
- Contrôleur d'entrée OpenShift

Avant de commencer

- Le requis "[contrôleur d'entrée](#)" doit déjà être déployé.
- Le "[classe d'entrée](#)" correspondant au contrôleur d'entrée doit déjà être créé.

Étapes pour l'entrée Istio

1. Configurer l'entrée Istio.



Cette procédure suppose que Istio est déployé à l'aide du profil de configuration par défaut.

2. Rassemblez ou créez le certificat et le fichier de clé privée souhaités pour la passerelle d'entrée.

Vous pouvez utiliser un certificat signé par une autorité de certification ou auto-signé. Le nom commun doit être l'adresse Astra (FQDN).

Exemple de commande :

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

3. Créez un secret `tls secret` name de type `kubernetes.io/tls` Pour une clé privée TLS et un certificat dans `istio-system` namespace Comme décrit dans les secrets TLS.

Exemple de commande :

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



Le nom du secret doit correspondre au `spec.tls.secretName` fourni dans `istio-ingress.yaml` fichier.

4. Déployer une ressource d'entrée dans le `netapp-acc` (ou nom personnalisé) de l'espace de noms utilisant le type de ressource `v1` pour un schéma (`istio-Ingress.yaml` est utilisé dans cet exemple) :

```

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. Appliquer les modifications :

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

6. Vérifier l'état de l'entrée :

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

Réponse :

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

7. Terminer l'installation du centre de contrôle Astra.

Étapes du contrôleur d'entrée Nginx

1. Créer un secret de type `kubernetes.io/tls` Pour une clé privée TLS et un certificat dans `netapp-acc` (ou espace de noms personnalisé) comme décrit dans "[Secrets TLS](#)".
2. Déployez une ressource entrée dans `netapp-acc` (ou nom personnalisé) de l'espace de noms utilisant le type de ressource `v1` pour un schéma (`nginx-Ingress.yaml` est utilisé dans cet exemple) :

```
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. Appliquer les modifications :

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



NetApp recommande d'installer le contrôleur nginx en tant que déploiement plutôt qu'en tant que `daemonSet`.

Étapes du contrôleur d'entrée OpenShift

1. Procurez-vous votre certificat et obtenez les fichiers de clé, de certificat et d'autorité de certification prêts à l'emploi par la route OpenShift.
2. Création de la route 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
```

Connectez-vous à l'interface utilisateur du centre de contrôle Astra

Après avoir installé Astra Control Center, vous modifierez le mot de passe de l'administrateur par défaut et vous connecterez au tableau de bord de l'interface utilisateur de Astra Control Center.

Étapes

1. Dans un navigateur, saisissez le nom de domaine complet (y compris le `https://` prefix) que vous avez utilisé dans `astraAddress` dans le `astra_control_center.yaml` CR quand [Vous avez installé Astra Control Center](#).
2. Acceptez les certificats auto-signés si vous y êtes invité.



Vous pouvez créer un certificat personnalisé après la connexion.

3. Dans la page de connexion à Astra Control Center, entrez la valeur que vous avez utilisée `email` dans `astra_control_center.yaml` CR quand [Vous avez installé Astra Control Center](#), suivi du mot de passe de configuration initiale (`ACC-[UUID]`).



Si vous saisissez trois fois un mot de passe incorrect, le compte admin est verrouillé pendant 15 minutes.

4. Sélectionnez **connexion**.
5. Modifiez le mot de passe lorsque vous y êtes invité.



S'il s'agit de votre première connexion et que vous oubliez le mot de passe et qu'aucun autre compte d'utilisateur administratif n'a encore été créé, contactez "[Support NetApp](#)" pour obtenir de l'aide sur la récupération des mots de

6. (Facultatif) supprimez le certificat TLS auto-signé existant et remplacez-le par un "[Certificat TLS personnalisé signé par une autorité de certification](#)".

Dépanner l'installation

Si l'un des services est dans `Error` état, vous pouvez inspecter les journaux. Recherchez les codes de réponse API dans la plage 400 à 500. Ceux-ci indiquent l'endroit où un échec s'est produit.

Options

- Pour inspecter les journaux de l'opérateur de l'Astra Control Center, entrez ce qui suit :

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

- Pour vérifier la sortie de l'Astra Control Center CR :

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

Et la suite

- (Facultatif) en fonction de votre environnement, effectuez l'installation complète après l'installation "[étapes de configuration](#)".
- Terminez le déploiement en effectuant le processus "[tâches de configuration](#)".

Configurez un gestionnaire de certificats externe

Si un gestionnaire de certificats existe déjà dans votre cluster Kubernetes, vous devez effectuer certaines étapes préalables afin qu'Astra Control Center n'installe pas son propre gestionnaire de certificats.

Étapes

1. Vérifiez qu'un gestionnaire de certificats est installé :

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

Exemple de réponse :

```
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. Créez une paire de certificats/clés pour le `astraAddress` FQDN :

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

Exemple de réponse :

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

3. Créez un secret avec des fichiers générés précédemment :

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

Exemple de réponse :

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

4. Créer un ClusterIssuer fichier qui est **exactement** le suivant mais qui comprend l'emplacement de l'espace de noms où votre cert-manager des pods sont installés :

```
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
```

Exemple de réponse :

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

5. Vérifiez que le ClusterIssuer s'est correctement installé. Ready doit être de True avant de pouvoir continuer :

```
kubectl get ClusterIssuer
```

Exemple de réponse :

NAME	READY	AGE
astra-ca-clusterissuer	True	9s

6. Complétez le "[Procédure d'installation d'Astra Control Center](#)". Il y a un "[Étape de configuration requise pour le groupe de centre de contrôle Astra YAML](#)" Dans lequel vous modifiez la valeur CRD pour indiquer que le gestionnaire de certificats est installé en externe. Vous devez effectuer cette étape pendant l'installation pour que le centre de contrôle Astra reconnaisse le responsable du certificat externe.

Installez Astra Control Center à l'aide d'OpenShift OperatorHub

Si vous utilisez Red Hat OpenShift, vous pouvez installer Astra Control Center à l'aide de l'opérateur certifié Red Hat. Utilisez cette procédure pour installer le centre de contrôle Astra à partir du "[Catalogue de l'écosystème Red Hat](#)" Ou utilisez Red Hat OpenShift Container Platform.

Une fois cette procédure terminée, vous devez revenir à la procédure d'installation pour terminer le "[les étapes restantes](#)" pour vérifier que l'installation a réussi et ouvrir une session.

Avant de commencer

- * Conditions préalables à l'environnement remplies* : "[Avant de commencer l'installation, préparez votre environnement pour le déploiement d'Astra Control Center](#)".
- **Opérateurs de grappe et services API sains** :
 - Depuis votre cluster OpenShift, assurez-vous que tous les opérateurs de clusters sont en état sain :

```
oc get clusteroperators
```

- Depuis votre cluster OpenShift, assurez-vous que tous les services d'API sont en état sain :

```
oc get apiservices
```

- **Adresse FQDN** : obtention d'une adresse FQDN pour Astra Control Center dans votre centre de données.
- **Autorisations OpenShift** : obtenez les autorisations nécessaires et l'accès à Red Hat OpenShift Container Platform pour effectuer les étapes d'installation décrites.
- **Cert Manager configuré** : si un cert Manager existe déjà dans le cluster, vous devez en effectuer certaines "[étapes préalables](#)" Pour qu'Astra Control Center n'installe pas son propre gestionnaire de certificat. Par défaut, Astra Control Center installe son propre gestionnaire de certificats lors de l'installation.
- **Contrôleur d'entrée Kubernetes** : si vous disposez d'un contrôleur d'entrée Kubernetes qui gère l'accès externe aux services, comme l'équilibrage de charge dans un cluster, vous devez le configurer pour l'utiliser avec Astra Control Center :
 - a. Créer l'espace de noms de l'opérateur :

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

- b. "[Terminez l'installation](#)" pour votre type de contrôleur d'entrée.

Étapes

- [Téléchargez et extrayez Astra Control Center](#)
- [Installez le plug-in NetApp Astra kubectl](#)
- [Ajoutez les images à votre registre local](#)

- [Recherchez la page d'installation de l'opérateur](#)
- [Poser l'opérateur](#)
- [Poser le centre de contrôle Astra](#)

Téléchargez et extrayez Astra Control Center

1. Accédez au "[Page de téléchargements d'Astra Control Center](#)" Sur le site de support NetApp.
2. Téléchargez le pack contenant Astra Control Center (`astra-control-center-[version].tar.gz`).
3. (Recommandé mais facultatif) Téléchargez le lot de certificats et de signatures pour Astra Control Center (`astra-control-center-certs-[version].tar.gz`) pour vérifier la signature du paquet :

```
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
```

La sortie s'affiche `Verified OK` une fois la vérification terminée.

4. Extraire les images du pack Astra Control Center :

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

Installez le plug-in NetApp Astra kubectl

Vous pouvez utiliser le plug-in de ligne de commande NetApp Astra kubectl pour envoyer les images vers un référentiel Docker local.

Avant de commencer

NetApp fournit des binaires de plug-ins pour différentes architectures CPU et systèmes d'exploitation. Avant d'effectuer cette tâche, vous devez savoir quelle unité centrale et quel système d'exploitation vous possédez.

Étapes

1. Répertoriez les binaires NetApp Astra kubectl disponibles et notez le nom du fichier dont vous avez besoin pour votre système d'exploitation et votre architecture de processeur :



La bibliothèque de plug-ins kubectl fait partie du bundle tar et est extraite dans le dossier `kubectl-astra`.

```
ls kubectl-astra/
```

2. Déplacez le bon binaire dans le chemin actuel et renommez-le `kubectl-astra`:

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

Ajoutez les images à votre registre local

1. Suivez la séquence d'étapes appropriée pour votre moteur de mise en conteneurs :

Docker

1. Accédez au répertoire racine du tarball. Vous devriez voir ce fichier et ce répertoire:

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

2. Envoyez les images du package dans le répertoire d'images Astra Control Center vers votre registre local. Effectuez les remplacements suivants avant d'exécuter le `push-images` commande :

- Remplacez `<BUNDLE_FILE>` par le nom du fichier bundle Astra Control (`acc.manifest.bundle.yaml`).
- Remplacer `<MY_FULL_REGISTRY_PATH>` par l'URL du référentiel Docker, par exemple `<a href="https://<docker-registry>" class="bare">https://<docker-registry>"`.
- Remplacez `<MY_REGISTRY_USER>` par le nom d'utilisateur.
- Remplacez `<MY_REGISTRY_TOKEN>` par un jeton autorisé pour le registre.

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

Podman

1. Accédez au répertoire racine du tarball. Vous devriez voir ce fichier et ce répertoire:

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

2. Connectez-vous à votre registre :

```
podman login <YOUR_REGISTRY>
```

3. Préparez et exécutez l'un des scripts suivants qui est personnalisé pour la version de Podman que vous utilisez. Remplacez `<MY_FULL_REGISTRY_PATH>` par l'URL de votre référentiel qui inclut tous les sous-répertoires.

```
<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

```



Le chemin d'accès à l'image que le script crée doit ressembler aux éléments suivants, selon la configuration de votre registre :

```

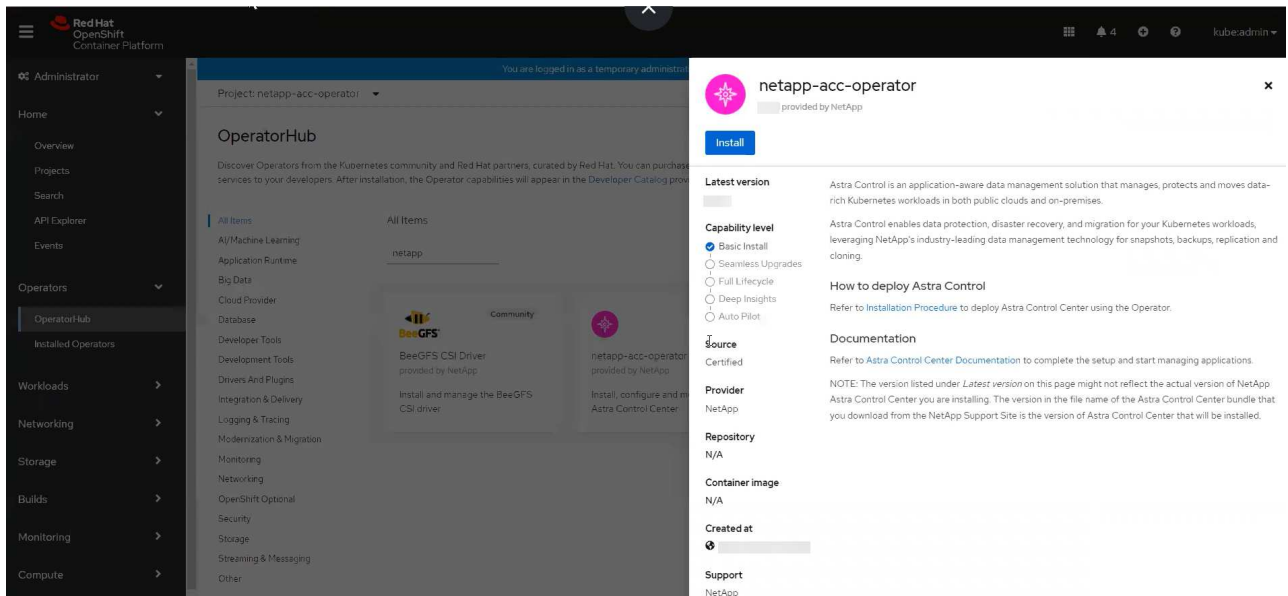
https://netappdownloads.jfrog.io/docker-astra-control-
prod/netapp/astra/acc/23.04.2-7/image:version

```

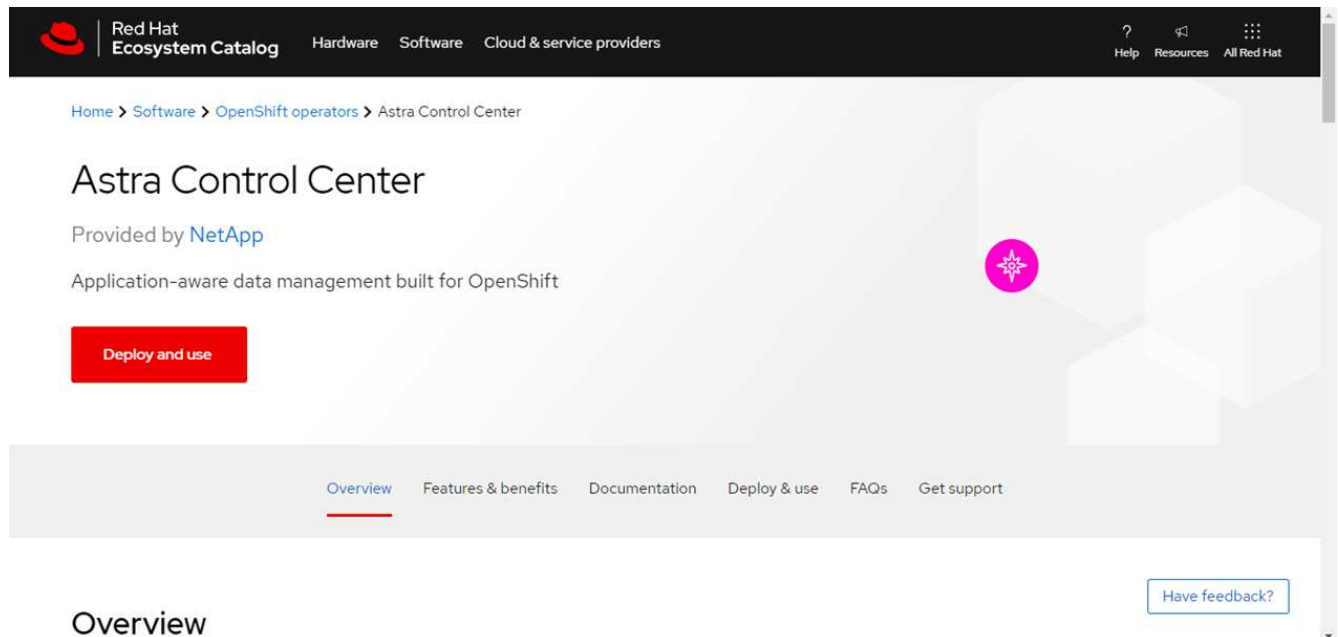
Recherchez la page d'installation de l'opérateur

1. Effectuez l'une des procédures suivantes pour accéder à la page d'installation de l'opérateur :

- Depuis la console Web Red Hat OpenShift :
 - i. Connectez-vous à l'interface utilisateur de OpenShift Container Platform.
 - ii. Dans le menu latéral, sélectionnez **Operators > OperatorHub**.
 - iii. Recherchez et sélectionnez l'opérateur NetApp Astra Control Center.



- À partir du catalogue de l'écosystème Red Hat :
 - i. Sélectionnez le centre de contrôle NetApp Astra "opérateur".
 - ii. Sélectionnez **deployer et utiliser**.



Poser l'opérateur

1. Complétez la page **Install Operator** et installez l'opérateur :



L'opérateur sera disponible dans tous les namespaces du cluster.

- a. Sélectionnez l'espace de noms de l'opérateur ou `netapp-acc-operator` l'espace de noms sera créé automatiquement dans le cadre de l'installation de l'opérateur.
- b. Sélectionnez une stratégie d'approbation manuelle ou automatique.



L'approbation manuelle est recommandée. Une seule instance d'opérateur doit s'exécuter par cluster.

- c. Sélectionnez **installer**.

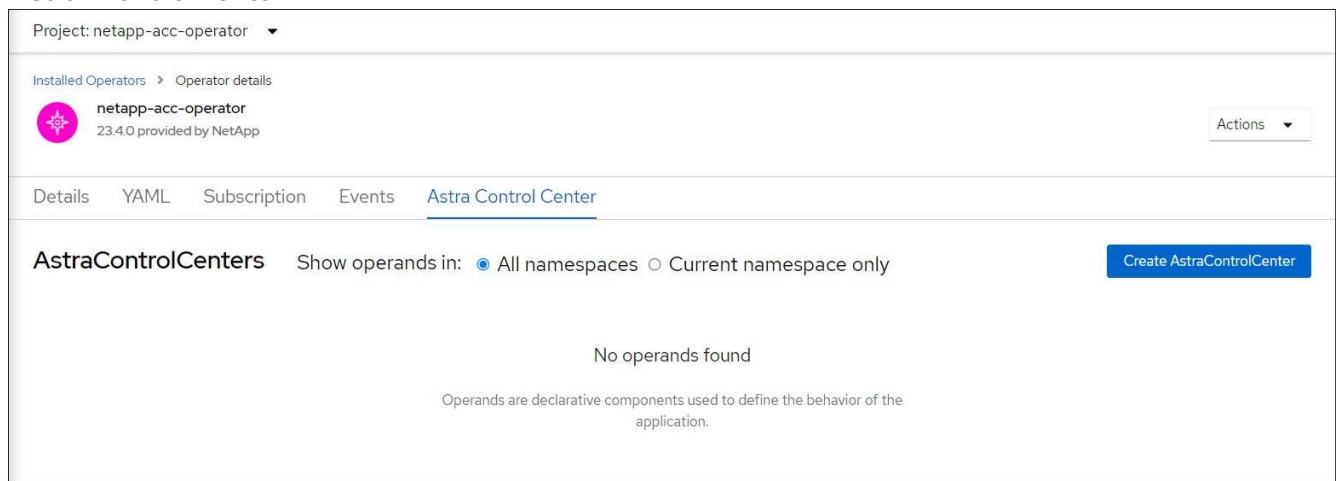


Si vous avez sélectionné une stratégie d'approbation manuelle, vous serez invité à approuver le plan d'installation manuelle pour cet opérateur.

2. Depuis la console, accéder au menu OperatorHub et vérifier que l'opérateur a bien installé.

Poser le centre de contrôle Astra

1. Dans la console de l'onglet **Astra Control Center** de l'opérateur Astra Control Center, sélectionnez **Create AstrakControlCenter**.



2. Complétez le `Create AstraControlCenter` champ de formulaire :
 - a. Conservez ou ajustez le nom du centre de contrôle Astra.
 - b. Ajouter des étiquettes pour le centre de contrôle Astra.
 - c. Activez ou désactivez Auto support. Il est recommandé de conserver la fonctionnalité Auto support.
 - d. Saisissez le nom de domaine complet ou l'adresse IP d'Astra Control Center. N'entrez pas `http://` ou `https://` dans le champ d'adresse.
 - e. Entrez la version d'Astra Control Center, par exemple 23.04.2-7.
 - f. Entrez un nom de compte, une adresse e-mail et un nom d'administrateur.
 - g. Choisir une règle de récupération de volume de `Retain`, `Recycle`, ou `Delete`. La valeur par défaut est `Retain`.
 - h. Sélectionnez la taille de l'échelle de l'installation.



Par défaut, Astra utilisera la haute disponibilité (HA) `scaleSize` de `Medium`, Qui déploie la plupart des services en haute disponibilité et déploie plusieurs répliques pour assurer la redondance. Avec `scaleSize` comme `Small`, Astra réduira le nombre de répliques pour tous les services, à l'exception des services essentiels, afin de réduire la consommation.

i. Sélectionnez le type d'entrée :

▪ **Generic** (`ingressType: "Generic"`) (Par défaut)

Utilisez cette option si vous avez un autre contrôleur d'entrée en service ou si vous préférez utiliser votre propre contrôleur d'entrée. Après le déploiement du centre de contrôle Astra, vous devez configurer le "contrôleur d'entrée" Pour exposer Astra Control Center avec une URL.

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

Utilisez cette option lorsque vous préférez ne pas configurer de contrôleur d'entrée. Ceci déploie le centre de contrôle Astra `traefik` Passerelle en tant que service de type Kubernetes « LoadBalancer ».

Le centre de contrôle Astra utilise un service de type « équilibreur de charge » (`svc/traefik` Dans l'espace de noms du centre de contrôle Astra), et exige qu'il se voit attribuer une adresse IP externe accessible. Si des équilibreurs de charge sont autorisés dans votre environnement et que vous n'en avez pas encore configuré, vous pouvez utiliser MetalLB ou un autre équilibreur de charge de service externe pour attribuer une adresse IP externe au service. Dans la configuration du serveur DNS interne, pointez le nom DNS choisi pour Astra Control Center vers l'adresse IP à équilibrage de charge.



Pour plus de détails sur le type de service « LoadBalancer » et Ingress, reportez-vous à la section "[De formation](#)".

- a. Dans **image Registry**, entrez le chemin d'accès au registre d'images du conteneur local. N'entrez pas `http://` ou `https://` dans le champ d'adresse.
- b. Si vous utilisez un registre d'images qui nécessite une authentification, saisissez le secret d'image.



Si vous utilisez un registre qui nécessite une authentification, [créez un secret sur le cluster](#).

- c. Entrez le prénom de l'administrateur.
- d. Configurer l'évolutivité des ressources.
- e. Indiquez la classe de stockage par défaut.



Si une classe de stockage par défaut est configurée, assurez-vous qu'elle est la seule classe de stockage qui possède l'annotation par défaut.

- f. Définissez les préférences de gestion de CRD.
3. Sélectionnez la vue YAML pour vérifier les paramètres sélectionnés.
 4. Sélectionnez `Create`.

Créer un secret de registre

Si vous utilisez un registre qui nécessite une authentification, créez un secret sur le cluster OpenShift et entrez le nom secret dans le `Create AstraControlCenter` champ de formulaire.

1. Créez un espace de noms pour l'opérateur du centre de contrôle Astra :

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

2. Créez un secret dans ce 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]
```



Astra Control prend uniquement en charge les secrets de registre Docker.

3. Renseignez les champs restants dans [Le champ de formulaire Create AstrakControlCenter](#).

Et la suite

Complétez le "[les étapes restantes](#)" Pour vérifier que le centre de contrôle Astra est correctement installé, configurez un contrôleur d'entrée (en option) et connectez-vous à l'interface utilisateur. De plus, vous devez effectuer cette opération "[tâches de configuration](#)" une fois l'installation terminée.

Installer le centre de contrôle Astra avec un système de stockage back-end Cloud Volumes ONTAP

Avec Astra Control Center, vous pouvez gérer les applications dans un environnement de cloud hybride avec des clusters Kubernetes et des instances Cloud Volumes ONTAP autogérés. Vous pouvez déployer Astra Control Center dans vos clusters Kubernetes sur site ou dans l'un des clusters Kubernetes autogéré dans l'environnement cloud.

Dans l'un de ces déploiements, vous pouvez effectuer des opérations de gestion des données d'application en utilisant Cloud Volumes ONTAP comme système back-end. Vous pouvez également configurer un compartiment S3 en tant que cible de sauvegarde.

Pour installer Astra Control Center dans Amazon Web Services (AWS), Google Cloud Platform (GCP) et Microsoft Azure avec un système back-end de stockage Cloud Volumes ONTAP, effectuez les opérations suivantes en fonction de votre environnement cloud.

- [Déploiement d'Astra Control Center dans Amazon Web Services](#)
- [Déployez Astra Control Center dans Google Cloud Platform](#)
- [Déploiement d'Astra Control Center dans Microsoft Azure](#)

Vous pouvez gérer vos applications dans des distributions avec des clusters Kubernetes autogérés, tels qu'OpenShift Container Platform (OCP). Seuls les clusters OCP autogérés sont validés pour le déploiement

d'Astra Control Center.

Déploiement d'Astra Control Center dans Amazon Web Services

Vous pouvez déployer Astra Control Center sur un cluster Kubernetes autogéré, hébergé dans un cloud public Amazon Web Services (AWS).

Ce dont vous avez besoin pour AWS

Avant de déployer Astra Control Center dans AWS, vous aurez besoin des éléments suivants :

- Licence Astra Control Center. Reportez-vous à la section "[Exigences de licence d'Astra Control Center](#)".
- "[Découvrez les exigences d'Astra Control Center](#)".
- Compte NetApp Cloud Central
- En cas d'utilisation des autorisations OCP, Red Hat OpenShift Container Platform (OCP) (au niveau de l'espace de noms pour créer des pods)
- Les identifiants AWS, l'ID d'accès et la clé secrète avec des autorisations qui vous permettent de créer des compartiments et des connecteurs
- Accès et connexion au registre d'instance de conteneur souple (ECR) du compte AWS
- Zone hébergée sur AWS et entrée route 53 nécessaires pour accéder à l'interface utilisateur de contrôle Astra

Exigences de l'environnement opérationnel pour AWS

Astra Control Center requiert l'environnement opérationnel suivant pour AWS :


- Red Hat OpenShift Container Platform 4.8



Assurez-vous que l'environnement d'exploitation que vous choisissez d'héberger est conforme aux exigences de base en matière de ressources décrites dans la documentation officielle de l'environnement.

Outre les exigences de l'environnement en matière de ressources, Astra Control Center requiert les ressources suivantes :

Composant	Conditions requises
Backend la capacité de stockage Cloud Volumes ONTAP	300 Go au moins disponibles
Nœuds workers (exigence AWS EC2)	Au moins 3 nœuds workers au total, avec 4 cœurs de vCPU et 12 Go de RAM chacun
Équilibrage de la charge	Type de service « LoadBalancer » disponible pour que le trafic d'entrée soit envoyé aux services du cluster d'environnement opérationnel
FQDN	Méthode permettant de pointer le FQDN de Astra Control Center vers l'adresse IP à charge équilibrée

Composant	Conditions requises
Astra Trident (installé dans le cadre de la découverte du cluster Kubernetes dans NetApp BlueXP, anciennement Cloud Manager)	Astra Trident 21.04 ou version ultérieure installé et configuré et NetApp ONTAP 9.5 ou version ultérieure en tant que système de stockage back-end
Registre d'images	<p>Vous devez disposer d'un registre privé existant, comme AWS Elastic Container Registry, auquel vous pouvez pousser les images de création Astra Control Center. Vous devez fournir l'URL du registre d'images où vous allez télécharger les images.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;">  <p>Le cluster hébergé par Astra Control Center et le cluster géré doivent avoir accès au même registre d'images pour pouvoir sauvegarder et restaurer des applications à l'aide de l'image Restic.</p> </div>
Configuration d'Astra Trident et ONTAP	<p>Avec Astra Control Center, il est nécessaire de créer une classe de stockage et de la définir comme classe de stockage par défaut. L'Astra Control Center prend en charge les classes de stockage Kubernetes ONTAP suivantes qui sont créées lorsque vous importez votre cluster Kubernetes dans NetApp BlueXP (anciennement Cloud Manager). Découvrez Astra Trident :</p> <ul style="list-style-type: none"> • <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>



De telles exigences supposent que Astra Control Center est la seule application qui s'exécute dans l'environnement opérationnel. Si l'environnement exécute des applications supplémentaires, ajustez ces exigences minimales en conséquence.



Le jeton de Registre AWS expire dans 12 heures. Après cela, vous devrez renouveler le code secret de Registre d'images Docker.

Présentation du déploiement pour AWS

Voici un aperçu du processus d'installation d'Astra Control Center pour AWS avec Cloud Volumes ONTAP en tant que système de stockage back-end.

Chacune de ces étapes est expliquée en détail ci-dessous.

1. [Assurez-vous que vous disposez de suffisamment d'autorisations IAM.](#)
2. [Installez un cluster Red Hat OpenShift sur AWS.](#)

3. [Configuration d'AWS.](#)
4. [Configuration de NetApp BlueXP pour AWS.](#)
5. [Installer Astra Control Center pour AWS.](#)

Assurez-vous que vous disposez de suffisamment d'autorisations IAM

Assurez-vous de disposer de suffisamment de rôles et d'autorisations IAM pour installer un cluster RedHat OpenShift et un connecteur NetApp BlueXP (anciennement Cloud Manager).

Voir ["Identifiants AWS initiaux"](#).

Installez un cluster Red Hat OpenShift sur AWS

Installez un cluster Red Hat OpenShift Container Platform sur AWS.

Pour obtenir des instructions d'installation, reportez-vous à la section ["Installation d'un cluster sur AWS dans OpenShift Container Platform"](#).

Configuration d'AWS

Configurez ensuite AWS pour créer un réseau virtuel, configurez les instances de calcul EC2, créez un compartiment AWS S3, créez un registre d'objets élastiques (ECR) pour héberger les images d'Astra Control Center et envoyez les images dans ce registre.

Suivez la documentation AWS pour suivre la procédure ci-dessous. Voir ["Documentation d'installation d'AWS"](#).

1. Créez un réseau virtuel AWS.
2. Vérifiez les instances de calcul EC2. Il peut s'agir d'un serveur bare Metal ou de machines virtuelles dans AWS.
3. Si le type d'instance ne correspond pas déjà aux exigences de ressources minimales Astra pour les nœuds maîtres et workers, modifiez le type d'instance dans AWS afin qu'il réponde aux exigences de l'Astra. Reportez-vous à la section ["Exigences du centre de contrôle Astra"](#).
4. Créez au moins un compartiment AWS S3 pour stocker vos sauvegardes.
5. Créez un registre AWS Elastic Container (ECR) pour héberger toutes les images ACC.



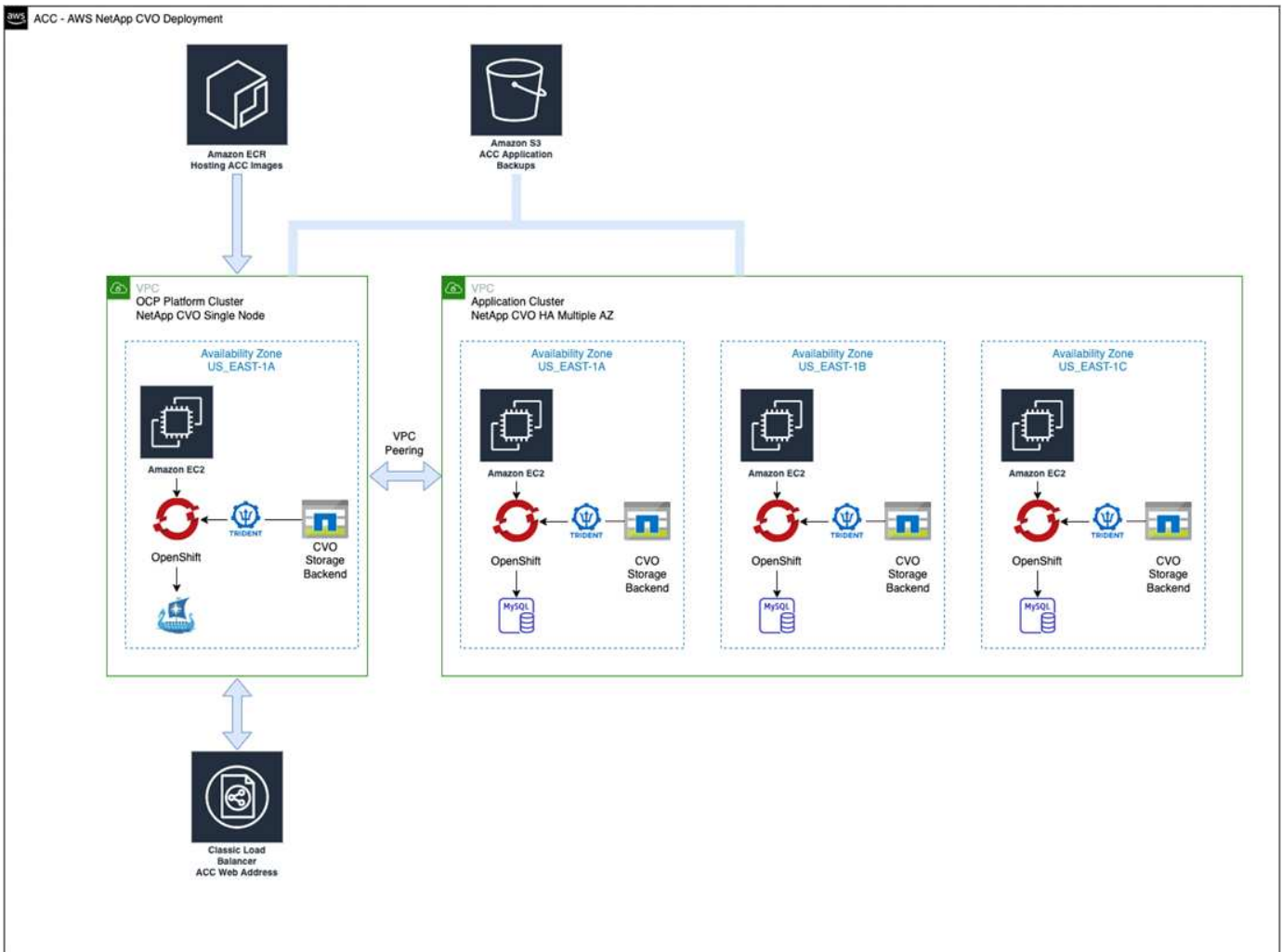
Si vous ne créez pas d'ECR, le centre de contrôle Astra ne peut pas accéder aux données de surveillance à partir d'un cluster contenant Cloud Volumes ONTAP avec un back-end AWS. Le problème survient lorsque le cluster que vous essayez de découvrir et de gérer à l'aide d'Astra Control Center n'a pas accès à AWS ECR.

6. Poussez les images ACC dans le registre défini.



Le token AWS Elastic Container Registry (ECR) expire au bout de 12 heures et provoque l'échec des opérations de clonage inter-cluster. Ce problème survient lors de la gestion d'un système back-end à partir d'Cloud Volumes ONTAP configuré pour AWS. Pour corriger ce problème, authentifiez-vous à nouveau avec l'ECR et générez un nouveau secret pour que les opérations de clonage puissent reprendre avec succès.

Voici un exemple de déploiement AWS :



Configuration de NetApp BlueXP pour AWS

Avec NetApp BlueXP (anciennement Cloud Manager), créez un espace de travail, ajoutez un connecteur à AWS, créez un environnement de travail et importez le cluster.

Suivez la documentation BlueXP pour effectuer les étapes suivantes. Voir les éléments suivants :

- ["Mise en route de Cloud Volumes ONTAP dans AWS"](#).
- ["Créez un connecteur dans AWS à l'aide de BlueXP"](#)

Étapes

1. Ajoutez vos informations d'identification à BlueXP.
2. Créez un espace de travail.
3. Ajoutez un connecteur pour AWS. Choisissez AWS en tant que fournisseur.
4. Créez un environnement de travail pour votre environnement cloud.
 - a. Emplacement : « Amazon Web Services (AWS) »
 - b. Type : « Cloud Volumes ONTAP HA »
5. Importer le cluster OpenShift Le cluster se connecte à l'environnement de travail que vous venez de créer.
 - a. Pour en savoir plus sur le cluster NetApp, sélectionnez **K8s > liste des clusters > Détails du cluster**.

- b. Dans le coin supérieur droit, notez la version d'Astra Trident.
- c. Notez les classes de stockage du cluster Cloud Volumes ONTAP indiquant NetApp comme provisionneur.

Cela importe votre cluster Red Hat OpenShift et lui attribue une classe de stockage par défaut. Vous sélectionnez la classe de stockage.

ASTRA Trident est automatiquement installé dans le cadre du processus d'importation et de découverte.

6. Noter tous les volumes et volumes persistants sur ce déploiement Cloud Volumes ONTAP



Cloud Volumes ONTAP peut fonctionner comme un seul nœud ou en mode haute disponibilité. Si la HA est activée, noter l'état de la HA et l'état du déploiement du nœud en cours dans AWS.

Installer Astra Control Center pour AWS

Respectez la norme "[Instructions d'installation du centre de contrôle Astra](#)".



AWS utilise le type de compartiment S3 générique.

Déployez Astra Control Center dans Google Cloud Platform

Vous pouvez déployer Astra Control Center sur un cluster Kubernetes autogéré, hébergé dans un cloud public Google Cloud Platform (GCP).

Éléments requis pour GCP

Avant de déployer Astra Control Center dans GCP, vous devez disposer des éléments suivants :


- Licence Astra Control Center. Reportez-vous à la section "[Exigences de licence d'Astra Control Center](#)".
- "[Découvrez les exigences d'Astra Control Center](#)".
- Compte NetApp Cloud Central
- Si vous utilisez OCP, Red Hat OpenShift Container Platform (OCP) 4.10
- En cas d'utilisation des autorisations OCP, Red Hat OpenShift Container Platform (OCP) (au niveau de l'espace de noms pour créer des pods)
- Compte de service GCP avec les autorisations qui vous permettent de créer des compartiments et des connecteurs

Conditions requises pour l'environnement opérationnel de GCP



Assurez-vous que l'environnement d'exploitation que vous choisissez d'héberger est conforme aux exigences de base en matière de ressources décrites dans la documentation officielle de l'environnement.

Outre les exigences de l'environnement en matière de ressources, Astra Control Center requiert les ressources suivantes :

Composant	Conditions requises
Backend la capacité de stockage Cloud Volumes ONTAP	300 Go au moins disponibles
Nœuds workers (exigences de calcul GCP)	Au moins 3 nœuds workers au total, avec 4 cœurs de vCPU et 12 Go de RAM chacun
Équilibrage de la charge	Type de service « LoadBalancer » disponible pour que le trafic d'entrée soit envoyé aux services du cluster d'environnement opérationnel
FQDN (ZONE DNS GCP)	Méthode permettant de pointer le FQDN de Astra Control Center vers l'adresse IP à charge équilibrée
Astra Trident (installé dans le cadre de la découverte du cluster Kubernetes dans NetApp BlueXP, anciennement Cloud Manager)	Astra Trident 21.04 ou version ultérieure installé et configuré et NetApp ONTAP 9.5 ou version ultérieure en tant que système de stockage back-end
Registre d'images	<p>Vous devez disposer d'un registre privé existant, tel que le registre de conteneurs Google, auquel vous pouvez pousser les images de création d'Astra Control Center. Vous devez fournir l'URL du registre d'images où vous allez télécharger les images.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Vous devez activer l'accès anonyme pour extraire les images Restic pour les sauvegardes. </div>
Configuration d'Astra Trident et ONTAP	<p>Avec Astra Control Center, il est nécessaire de créer une classe de stockage et de la définir comme classe de stockage par défaut. Astra Control Center prend en charge les classes de stockage Kubernetes ONTAP suivantes qui sont créées lorsque vous importez votre cluster Kubernetes dans NetApp BlueXP. Découvrez Astra Trident :</p> <ul style="list-style-type: none"> • <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>



De telles exigences supposent que Astra Control Center est la seule application qui s'exécute dans l'environnement opérationnel. Si l'environnement exécute des applications supplémentaires, ajustez ces exigences minimales en conséquence.

Présentation du déploiement pour GCP

Voici un aperçu du processus d'installation d'Astra Control Center sur un cluster OCP autogéré dans GCP

avec Cloud Volumes ONTAP comme système de stockage principal.

Chacune de ces étapes est expliquée en détail ci-dessous.

1. [Installez un cluster RedHat OpenShift sur GCP.](#)
2. [Création d'un projet GCP et d'un cloud privé virtuel.](#)
3. [Assurez-vous que vous disposez de suffisamment d'autorisations IAM.](#)
4. [Configurer GCP.](#)
5. [Configurez NetApp BlueXP pour GCP.](#)
6. [Installez Astra Control Center pour GCP.](#)

Installez un cluster RedHat OpenShift sur GCP

La première étape consiste à installer un cluster Red Hat OpenShift sur GCP.

Pour les instructions d'installation, reportez-vous aux sections suivantes :

- ["Installation d'un cluster OpenShift dans GCP"](#)
- ["Création d'un compte de service GCP"](#)

Création d'un projet GCP et d'un cloud privé virtuel

Créez au moins un projet GCP et un cloud privé virtuel (VPC).



OpenShift peut créer ses propres groupes de ressources. En plus de ces VPC, vous devez également définir un VPC GCP. Voir la documentation OpenShift.

Vous pouvez créer un groupe de ressources de cluster de plate-forme et un groupe de ressources de cluster OpenShift d'application cible.

Assurez-vous que vous disposez de suffisamment d'autorisations IAM

Assurez-vous de disposer de suffisamment de rôles et d'autorisations IAM pour installer un cluster RedHat OpenShift et un connecteur NetApp BlueXP (anciennement Cloud Manager).

Voir ["Identifiants et autorisations GCP initiaux"](#).

Configurer GCP

Configurez ensuite GCP pour créer un VPC, configurez des instances de calcul, créez un stockage objet Google Cloud, créez un registre de conteneurs Google pour héberger les images d'Astra Control Center et envoyez les images vers ce registre.

Suivez la documentation GCP pour effectuer les étapes suivantes. Voir installation du cluster OpenShift dans GCP.

1. Créez un projet GCP et un VPC dans le GCP que vous prévoyez d'utiliser pour le cluster OCP avec le backend CVO.
2. Vérifiez les instances de calcul. Il peut s'agir d'un serveur bare Metal ou de machines virtuelles dans GCP.
3. Si le type d'instance ne correspond pas déjà aux exigences minimales de ressources d'Astra pour les nœuds maîtres et workers, modifiez le type d'instance dans GCP pour répondre aux exigences d'Astra.

Reportez-vous à la section ["Exigences du centre de contrôle Astra"](#).

4. Créez au moins un compartiment de stockage cloud GCP pour stocker vos sauvegardes.
5. Créez un secret, requis pour l'accès au compartiment.
6. Créez un registre de conteneurs Google pour héberger toutes les images du centre de contrôle Astra.
7. Configurez l'accès du registre de conteneurs Google pour le transfert/transfert de Docker pour toutes les images du centre de contrôle Astra.

Exemple : les images ACC peuvent être transmises à ce registre en entrant le script suivant :

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

Ce script nécessite un fichier manifeste Astra Control Center et votre emplacement dans le registre d'images Google.

Exemple :

```
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. Configurer les zones DNS.

Configurez NetApp BlueXP pour GCP

À l'aide de NetApp BlueXP (anciennement Cloud Manager), créez un espace de travail, ajoutez un connecteur à GCP, créez un environnement de travail et importez le cluster.

Suivez la documentation BlueXP pour effectuer les étapes suivantes. Voir ["Mise en route de Cloud Volumes ONTAP dans GCP"](#).

Avant de commencer

- Accès au compte de services GCP avec les autorisations IAM et les rôles requis

Étapes

1. Ajoutez vos informations d'identification à BlueXP. Voir ["Ajout de comptes GCP"](#).
2. Ajouter un connecteur pour GCP.

- a. Choisissez GCP comme fournisseur.
 - b. Entrez les identifiants GCP. Voir "[Création d'un connecteur dans GCP à partir de BlueXP](#)".
 - c. S'assurer que le connecteur est en marche et basculer vers ce connecteur.
3. Créez un environnement de travail pour votre environnement cloud.
 - a. Emplacement : « GCP »
 - b. Type : « Cloud Volumes ONTAP HA »
 4. Importer le cluster OpenShift Le cluster se connecte à l'environnement de travail que vous venez de créer.
 - a. Pour en savoir plus sur le cluster NetApp, sélectionnez **K8s > liste des clusters > Détails du cluster**.
 - b. Notez la version Trident dans le coin supérieur droit.
 - c. Notez les classes de stockage du cluster Cloud Volumes ONTAP indiquant « NetApp » comme provisionneur.

Cela importe votre cluster Red Hat OpenShift et lui attribue une classe de stockage par défaut. Vous sélectionnez la classe de stockage. ASTRA Trident est automatiquement installé dans le cadre du processus d'importation et de découverte.

5. Noter tous les volumes et volumes persistants sur ce déploiement Cloud Volumes ONTAP



Cloud Volumes ONTAP peut fonctionner comme un seul nœud ou en haute disponibilité. Si la haute disponibilité est activée, notez l'état de la haute disponibilité et l'état du déploiement des nœuds exécutés dans GCP.

Installez Astra Control Center pour GCP

Respectez la norme "[Instructions d'installation du centre de contrôle Astra](#)".



GCP utilise le type de compartiment S3 générique.

1. Générez le secret Docker pour extraire des images pour l'installation du centre de contrôle Astra :

```
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
```

Déploiement d'Astra Control Center dans Microsoft Azure

Vous pouvez déployer Astra Control Center sur un cluster Kubernetes autogéré, hébergé dans un cloud public Microsoft Azure.

Ce dont vous avez besoin pour Azure

Avant de déployer Astra Control Center dans Azure, vous aurez besoin des éléments suivants :

- Licence Astra Control Center. Reportez-vous à la section "[Exigences de licence d'Astra Control Center](#)".
- "[Découvrez les exigences d'Astra Control Center](#)".


- Compte NetApp Cloud Central
- Si vous utilisez OCP, Red Hat OpenShift Container Platform (OCP) 4.8
- En cas d'utilisation des autorisations OCP, Red Hat OpenShift Container Platform (OCP) (au niveau de l'espace de noms pour créer des pods)
- Les identifiants Azure avec autorisations qui vous permettent de créer des compartiments et des connecteurs

Exigences de l'environnement opérationnel pour Azure

Assurez-vous que l'environnement d'exploitation que vous choisissez d'héberger est conforme aux exigences de base en matière de ressources décrites dans la documentation officielle de l'environnement.

Outre les exigences de l'environnement en matière de ressources, Astra Control Center requiert les ressources suivantes :

Reportez-vous à la section "[Exigences relatives à l'environnement opérationnel d'Astra Control Center](#)".

Composant	Conditions requises
Backend la capacité de stockage Cloud Volumes ONTAP	300 Go au moins disponibles
Nœuds worker (exigences de calcul Azure)	Au moins 3 nœuds workers au total, avec 4 cœurs de vCPU et 12 Go de RAM chacun
Équilibrage de la charge	Type de service « LoadBalancer » disponible pour que le trafic d'entrée soit envoyé aux services du cluster d'environnement opérationnel
FQDN (zone Azure DNS)	Méthode permettant de pointer le FQDN de Astra Control Center vers l'adresse IP à charge équilibrée
Astra Trident (installé dans le cadre de la découverte du cluster Kubernetes dans NetApp BlueXP)	Astra Trident 21.04 ou version ultérieure installé et configuré et NetApp ONTAP version 9.5 ou ultérieure sera utilisé comme système de stockage back-end
Registre d'images	<p>Vous devez disposer d'un registre privé existant, tel que le registre de conteneur Azure (ACR), auquel vous pouvez pousser les images de création d'Astra Control Center. Vous devez fournir l'URL du registre d'images où vous allez télécharger les images.</p> <div style="display: flex; align-items: center;">  <p>Vous devez activer l'accès anonyme pour extraire les images Restic pour les sauvegardes.</p> </div>

Composant	Conditions requises
Configuration d'Astra Trident et ONTAP	<p>Avec Astra Control Center, il est nécessaire de créer une classe de stockage et de la définir comme classe de stockage par défaut. Astra Control Center prend en charge les classes de stockage Kubernetes ONTAP suivantes qui sont créées lorsque vous importez votre cluster Kubernetes dans NetApp BlueXP. Découvrez Astra Trident :</p> <ul style="list-style-type: none"> • <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>



De telles exigences supposent que Astra Control Center est la seule application qui s'exécute dans l'environnement opérationnel. Si l'environnement exécute des applications supplémentaires, ajustez ces exigences minimales en conséquence.

Présentation du déploiement pour Azure

Voici un aperçu du processus d'installation d'Astra Control Center pour Azure.

Chacune de ces étapes est expliquée en détail ci-dessous.

1. [Installez un cluster Red Hat OpenShift sur Azure.](#)
2. [Créez des groupes de ressources Azure.](#)
3. [Assurez-vous que vous disposez de suffisamment d'autorisations IAM.](#)
4. [Configurez Azure.](#)
5. [Configuration de NetApp BlueXP \(anciennement Cloud Manager\) pour Azure.](#)
6. [Installer et configurer Astra Control Center pour Azure.](#)

Installez un cluster Red Hat OpenShift sur Azure

La première étape consiste à installer un cluster Red Hat OpenShift sur Azure.

Pour les instructions d'installation, reportez-vous aux sections suivantes :

- ["Installation du cluster OpenShift sur Azure"](#).
- ["Installation d'un compte Azure"](#).

Créez des groupes de ressources Azure

Créez au moins un groupe de ressources Azure.



OpenShift peut créer ses propres groupes de ressources. En plus de ces groupes, vous devez également définir des groupes de ressources Azure. Voir la documentation OpenShift.

Vous pouvez créer un groupe de ressources de cluster de plate-forme et un groupe de ressources de cluster OpenShift d'application cible.

Assurez-vous que vous disposez de suffisamment d'autorisations IAM

Assurez-vous de disposer de suffisamment de rôles et d'autorisations IAM pour installer un cluster RedHat OpenShift et un connecteur NetApp BlueXP.

Voir "[Identifiants et autorisations Azure](#)".

Configurez Azure

Configurez ensuite Azure pour créer un réseau virtuel, configurez des instances de calcul, créez un conteneur Azure Blob Container Register, créez un ACR (Azure Container Register) pour héberger les images d'Astra Control Center et envoyez les images dans ce registre.

Suivez la documentation Azure pour suivre les étapes ci-dessous. Voir "[Installation du cluster OpenShift sur Azure](#)".

1. Créez un réseau virtuel Azure.
2. Vérifiez les instances de calcul. Il peut s'agir d'un serveur bare Metal ou de machines virtuelles dans Azure.
3. Si le type d'instance ne correspond pas déjà aux exigences de ressources minimales Astra pour les nœuds maîtres et workers, modifiez le type d'instance dans Azure afin qu'il réponde aux exigences de l'Astra. Reportez-vous à la section "[Exigences du centre de contrôle Astra](#)".
4. Créez au moins un conteneur Azure Blob pour stocker vos sauvegardes.
5. Créez un compte de stockage. Vous aurez besoin d'un compte de stockage pour créer un conteneur à utiliser comme compartiment dans Astra Control Center.
6. Créez un secret, requis pour l'accès au compartiment.
7. Créez un registre de conteneurs Azure (ACR) pour héberger toutes les images du centre de contrôle Astra.
8. Configurez l'accès ACR pour Docker pousser/extraire toutes les images du centre de contrôle Astra.
9. Envoyez les images ACC dans ce registre en entrant le script suivant :

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

Exemple :

```
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. Configurer les zones DNS.

Configuration de NetApp BlueXP (anciennement Cloud Manager) pour Azure

À l'aide de BlueXP (anciennement Cloud Manager), créez un espace de travail, ajoutez un connecteur à Azure, créez un environnement de travail et importez le cluster.

Suivez la documentation BlueXP pour effectuer les étapes suivantes. Voir "[Mise en route de BlueXP dans Azure](#)".

Avant de commencer

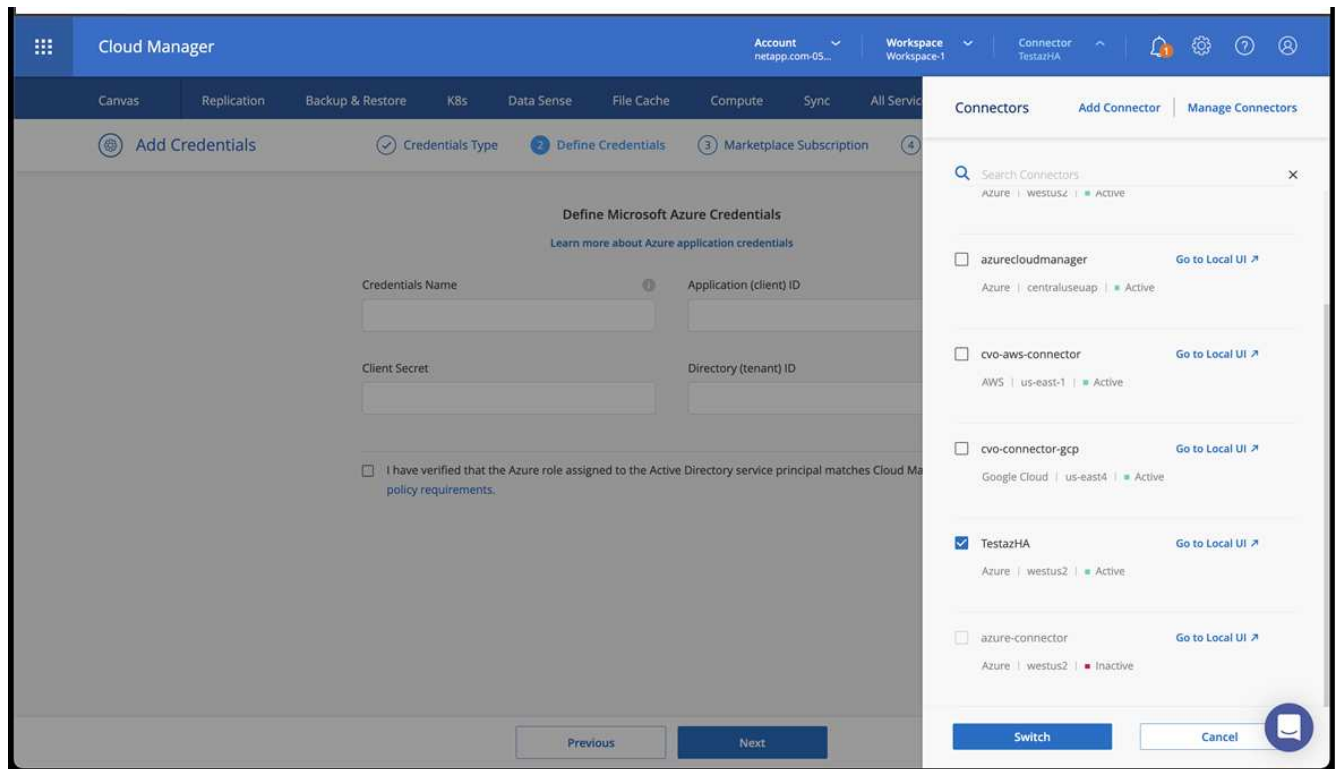
Accès au compte Azure avec les autorisations IAM et les rôles requis

Étapes

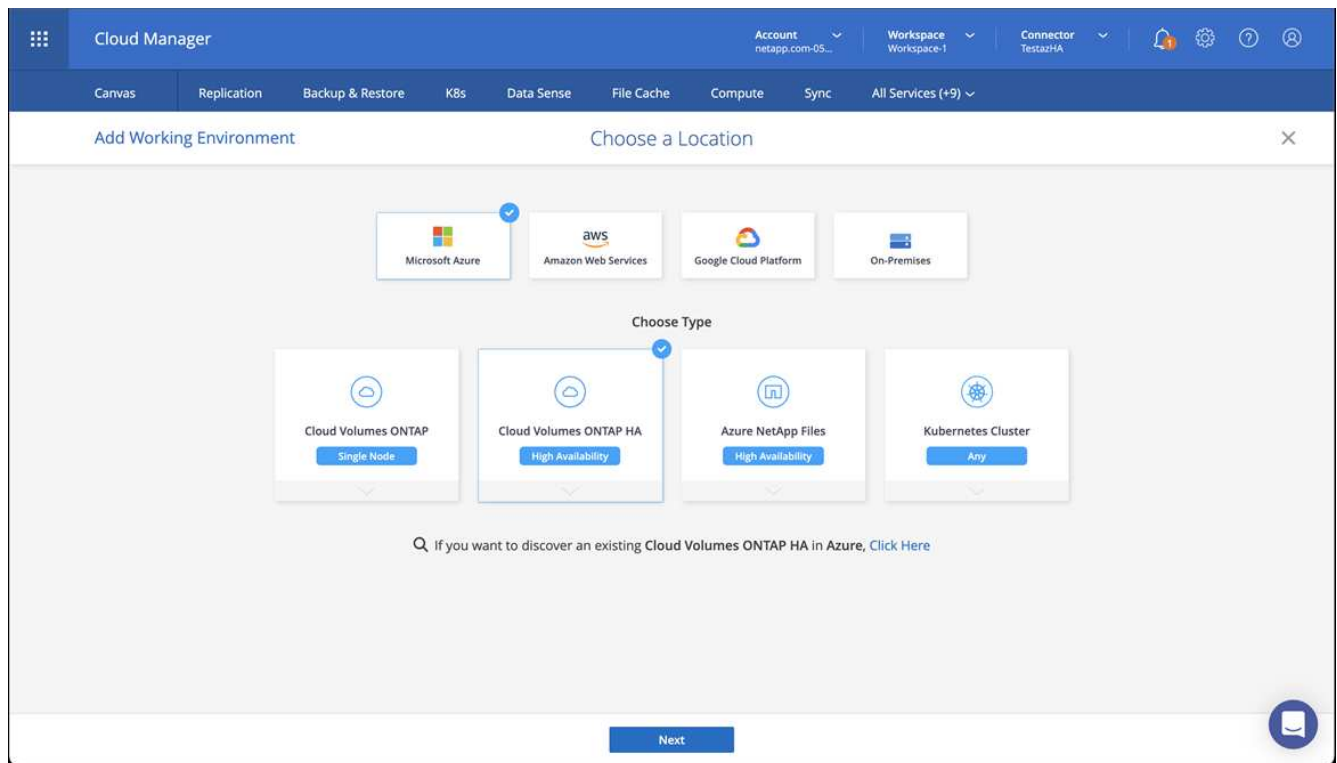
1. Ajoutez vos informations d'identification à BlueXP.
2. Ajoutez un connecteur pour Azure. Voir "[Politiques BlueXP](#)".
 - a. Choisissez **Azure** comme fournisseur.
 - b. Vous pouvez entrer les identifiants Azure, notamment l'ID de l'application, le secret client et l'ID du répertoire (locataire).

Voir "[Création d'un connecteur dans Azure à partir de BlueXP](#)".

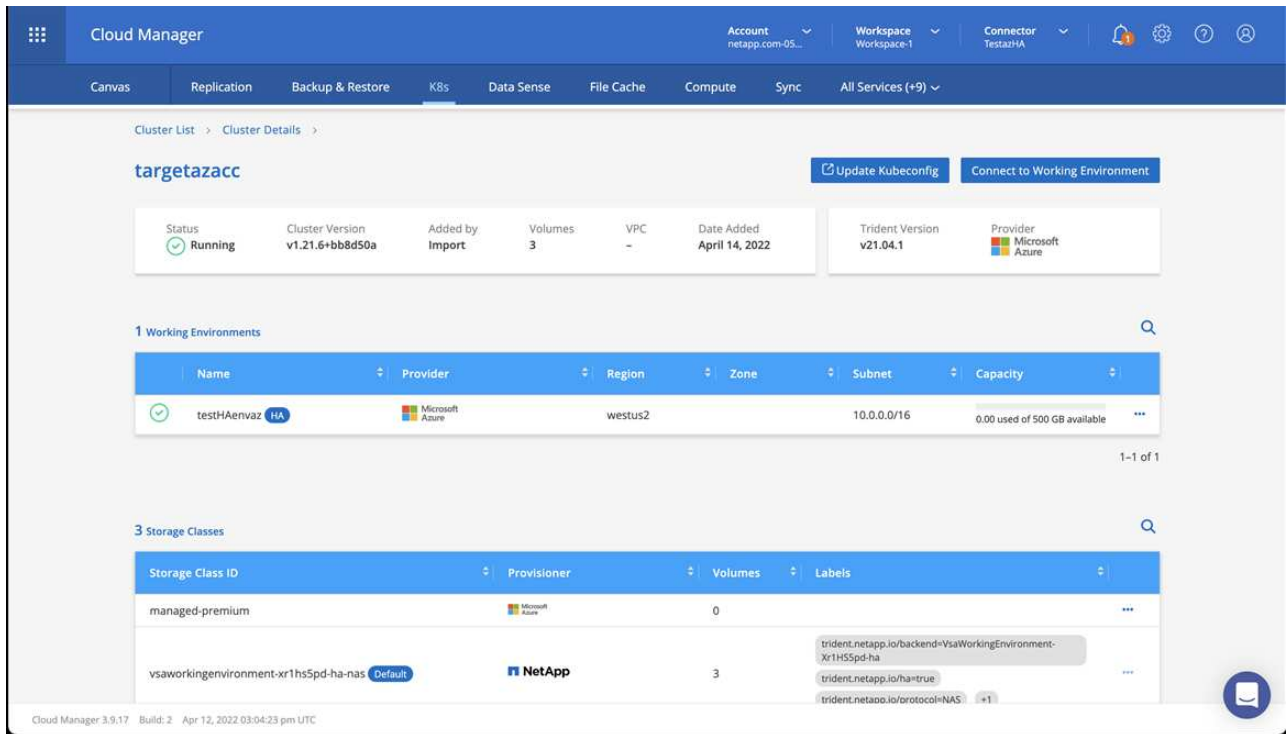
3. S'assurer que le connecteur est en marche et basculer vers ce connecteur.



4. Créez un environnement de travail pour votre environnement cloud.
 - a. Emplacement : « Microsoft Azure ».
 - b. Type : « Cloud Volumes ONTAP HA ».



5. Importer le cluster OpenShift Le cluster se connecte à l'environnement de travail que vous venez de créer.
 - a. Pour en savoir plus sur le cluster NetApp, sélectionnez **K8s > liste des clusters > Détails du cluster**.



- b. Dans le coin supérieur droit, notez la version d'Astra Trident.
- c. Notez les classes de stockage du cluster Cloud Volumes ONTAP indiquant NetApp comme provisionneur.

Cela importe votre cluster Red Hat OpenShift et attribue une classe de stockage par défaut. Vous sélectionnez la classe de stockage.

ASTRA Trident est automatiquement installé dans le cadre du processus d'importation et de découverte.

6. Noter tous les volumes et volumes persistants sur ce déploiement Cloud Volumes ONTAP
7. Cloud Volumes ONTAP peut fonctionner comme un seul nœud ou en mode haute disponibilité. Si la HA est activée, noter l'état de la HA et l'état du déploiement du nœud en cours d'exécution dans Azure.

Installer et configurer Astra Control Center pour Azure

Installer le centre de contrôle Astra de série "[instructions d'installation](#)".

Avec Astra Control Center, ajoutez un compartiment Azure. Reportez-vous à la section "[Configurer le centre de contrôle Astra et ajouter des seaux](#)".

Configurer le centre de contrôle Astra après l'installation

En fonction de votre environnement, une configuration supplémentaire peut être nécessaire après l'installation d'Astra Control Center.

Supprimer les limitations de ressources

Certains environnements utilisent les objets ResourceQuotas et LimitRanges pour empêcher les ressources d'un namespace de consommer l'ensemble des CPU et de la mémoire disponibles sur le cluster. Le centre de contrôle Astra ne fixe pas de limites maximales, il ne sera donc pas conforme à ces ressources. Si votre environnement est configuré de cette façon, vous devez supprimer ces ressources des espaces de noms où

vous prévoyez d'installer Astra Control Center.

Vous pouvez suivre les étapes suivantes pour récupérer et supprimer ces quotas et ces limites. Dans ces exemples, la sortie de la commande est affichée immédiatement après la commande.

Étapes

1. Obtenez les quotas de ressources dans `netapp-acc` (ou nom-personnalisé) espace de noms :

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

Réponse :

```
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. Supprimez tous les quotas de ressources par nom :

```
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. Obtenez les limites de la `netapp-acc` (ou nom-personnalisé) espace de noms :

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

Réponse :

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


4. Supprimez les plages de limite par nom :

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

Activer la communication réseau entre les espaces de noms

Certains environnements utilisent les constructions NetworkPolicy pour limiter le trafic entre les espaces de noms. L'opérateur du centre de contrôle Astra et le centre de contrôle Astra sont dans des espaces de noms différents. Les services de ces différents espaces de noms doivent être capables de communiquer les uns avec les autres. Pour activer cette communication, procédez comme suit.

Étapes

1. Supprimez toutes les ressources NetworkPolicy qui existent dans l'espace de noms Astra Control Center :

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

2. Pour chaque objet NetworkPolicy renvoyé par la commande précédente, utilisez la commande suivante pour le supprimer. Remplacez [NOM_OBJET] par le nom de l'objet renvoyé :

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

3. Appliquez le fichier de ressources suivant pour configurer le `acc-avp-network-policy` Objet pour permettre aux services de plug-in Astra de faire des demandes aux services Astra Control Center. Remplacez entre parenthèses <> par les informations relatives à votre environnement :

```
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. Appliquez le fichier de ressources suivant pour configurer le `acc-operator-network-policy` Objet permettant à l'opérateur du centre de contrôle Astra de communiquer avec les services du centre de contrôle Astra. Remplacez entre parenthèses `<>` par les informations relatives à votre environnement :

```
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
```

Ajouter un certificat TLS personnalisé

Par défaut, Astra Control Center utilise un certificat TLS auto-signé pour le trafic du contrôleur d'entrée (uniquement dans certaines configurations) et l'authentification de l'interface utilisateur Web avec des navigateurs Web. Vous pouvez supprimer le certificat TLS auto-signé existant et le remplacer par un certificat TLS signé par une autorité de certification (AC).



Le certificat auto-signé par défaut est utilisé pour deux types de connexions :

- Connexions HTTPS à l'interface utilisateur Web Astra Control Center
- Entrée du trafic du contrôleur (uniquement si le `ingressType: "AccTraefik"` la propriété a été définie dans `astra_control_center.yaml` Fichier lors de l'installation d'Astra Control Center)

Le remplacement du certificat TLS par défaut remplace le certificat utilisé pour l'authentification pour ces connexions.

Avant de commencer

- Cluster Kubernetes avec Astra Control Center installé
- Accès administratif à un shell de commande sur le cluster à exécuter `kubectl` commandes
- Clé privée et fichiers de certificat de l'autorité de certification

Supprimez le certificat auto-signé

Supprimez le certificat TLS auto-signé existant.

1. Avec SSH, connectez-vous au cluster Kubernetes qui héberge Astra Control Center en tant qu'utilisateur administratif.
2. Recherchez le code secret TLS associé au certificat en cours à l'aide de la commande suivante, remplacement <ACC-deployment-namespace> Avec l'espace de noms de déploiement d'Astra Control Center :

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

3. Supprimez le certificat et le secret actuellement installés à l'aide des commandes suivantes :

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

Ajoutez un nouveau certificat à l'aide de la ligne de commande

Ajoutez un nouveau certificat TLS signé par une autorité de certification.

1. Utilisez la commande suivante pour créer le nouveau secret TLS avec la clé privée et les fichiers de certificat de l'autorité de certification, en remplaçant les arguments entre parenthèses <> par les informations appropriées :

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

2. Utilisez la commande et l'exemple suivants pour modifier le fichier CRD (Custom Resource Definition) du cluster et modifier `spec.selfSigned` valeur à `spec.ca.secretName` Pour consulter le secret TLS créé précédemment :

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

#spec:
#  selfSigned: {}

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

3. Utilisez la commande suivante et exemple de résultat pour vérifier que les modifications sont correctes et le cluster est prêt à valider les certificats, en remplaçant <ACC-deployment-namespace> Avec l'espace de noms de déploiement d'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. Créer le `certificate.yaml` fichier avec l'exemple suivant, en remplaçant les valeurs de paramètre fictif entre parenthèses `<>` par les informations appropriées :

```
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. Créez le certificat à l'aide de la commande suivante :

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

6. À l'aide de la commande et de l'exemple de sortie suivants, vérifiez que le certificat a été créé correctement et avec les arguments que vous avez spécifiés lors de la création (tels que le nom, la durée, la date limite de renouvellement et les noms 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. Modifiez l'option Ingress CRD TLS pour pointer vers votre nouveau secret de certificat à l'aide de la commande suivante et de l'exemple, en remplaçant les valeurs de paramètre fictif entre parenthèses <> par les informations appropriées :

```
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. À l'aide d'un navigateur Web, accédez à l'adresse IP de déploiement d'Astra Control Center.
9. Vérifiez que les détails du certificat correspondent aux détails du certificat que vous avez installé.
10. Exportez le certificat et importez le résultat dans le gestionnaire de certificats de votre navigateur Web.

Informations sur le copyright

Copyright © 2023 NetApp, Inc. Tous droits réservés. Imprimé aux États-Unis. Aucune partie de ce document protégé par copyright ne peut être reproduite sous quelque forme que ce soit ou selon quelque méthode que ce soit (graphique, électronique ou mécanique, notamment par photocopie, enregistrement ou stockage dans un système de récupération électronique) sans l'autorisation écrite préalable du détenteur du droit de copyright.

Les logiciels dérivés des éléments NetApp protégés par copyright sont soumis à la licence et à l'avis de non-responsabilité suivants :

CE LOGICIEL EST FOURNI PAR NETAPP « EN L'ÉTAT » ET SANS GARANTIES EXPRESSES OU TACITES, Y COMPRIS LES GARANTIES TACITES DE QUALITÉ MARCHANDE ET D'ADÉQUATION À UN USAGE PARTICULIER, QUI SONT EXCLUES PAR LES PRÉSENTES. EN AUCUN CAS NETAPP NE SERA TENU POUR RESPONSABLE DE DOMMAGES DIRECTS, INDIRECTS, ACCESSOIRES, PARTICULIERS OU EXEMPLAIRES (Y COMPRIS L'ACHAT DE BIENS ET DE SERVICES DE SUBSTITUTION, LA PERTE DE JOUISSANCE, DE DONNÉES OU DE PROFITS, OU L'INTERRUPTION D'ACTIVITÉ), QUELLES QU'EN SOIENT LA CAUSE ET LA DOCTRINE DE RESPONSABILITÉ, QU'IL S'AGISSE DE RESPONSABILITÉ CONTRACTUELLE, STRICTE OU DÉLICTEUELLE (Y COMPRIS LA NÉGLIGENCE OU AUTRE) DÉCOULANT DE L'UTILISATION DE CE LOGICIEL, MÊME SI LA SOCIÉTÉ A ÉTÉ INFORMÉE DE LA POSSIBILITÉ DE TELS DOMMAGES.

NetApp se réserve le droit de modifier les produits décrits dans le présent document à tout moment et sans préavis. NetApp décline toute responsabilité découlant de l'utilisation des produits décrits dans le présent document, sauf accord explicite écrit de NetApp. L'utilisation ou l'achat de ce produit ne concède pas de licence dans le cadre de droits de brevet, de droits de marque commerciale ou de tout autre droit de propriété intellectuelle de NetApp.

Le produit décrit dans ce manuel peut être protégé par un ou plusieurs brevets américains, étrangers ou par une demande en attente.

LÉGENDE DE RESTRICTION DES DROITS : L'utilisation, la duplication ou la divulgation par le gouvernement sont sujettes aux restrictions énoncées dans le sous-paragraphe (b)(3) de la clause Rights in Technical Data-Noncommercial Items du DFARS 252.227-7013 (février 2014) et du FAR 52.227-19 (décembre 2007).

Les données contenues dans les présentes se rapportent à un produit et/ou service commercial (tel que défini par la clause FAR 2.101). Il s'agit de données propriétaires de NetApp, Inc. Toutes les données techniques et tous les logiciels fournis par NetApp en vertu du présent Accord sont à caractère commercial et ont été exclusivement développés à l'aide de fonds privés. Le gouvernement des États-Unis dispose d'une licence limitée irrévocable, non exclusive, non cessible, non transférable et mondiale. Cette licence lui permet d'utiliser uniquement les données relatives au contrat du gouvernement des États-Unis d'après lequel les données lui ont été fournies ou celles qui sont nécessaires à son exécution. Sauf dispositions contraires énoncées dans les présentes, l'utilisation, la divulgation, la reproduction, la modification, l'exécution, l'affichage des données sont interdits sans avoir obtenu le consentement écrit préalable de NetApp, Inc. Les droits de licences du Département de la Défense du gouvernement des États-Unis se limitent aux droits identifiés par la clause 252.227-7015(b) du DFARS (février 2014).

Informations sur les marques commerciales

NETAPP, le logo NETAPP et les marques citées sur le site <http://www.netapp.com/TM> sont des marques déposées ou des marques commerciales de NetApp, Inc. Les autres noms de marques et de produits sont des marques commerciales de leurs propriétaires respectifs.