Poser le centre de contrôle Astra

Astra Control Center

NetApp
August 11, 2025

This PDF was generated from https://docs.netapp.com/fr-fr/astra-control-center-2307/get-started/cert-manager-prereqs.html on August 11, 2025. Always check docs.netapp.com for the latest.

Sommaire

Installer le centre de contrôle Astra en suivant la procédure standard . . . 1
- Téléchargez et extrayez Astra Control Center . . . 2
- Installez le plug-in NetApp Astra kubectl . . . 2
- Ajoutez les images à votre registre local . . . 3
- Configurez l’espace de noms et le secret pour les registres avec les exigences d’authentification . . . 6
- Poser le conducteur du centre de commande Astra . . . 7
- Configurer le centre de contrôle Astra . . . 11
- Installation complète du centre de contrôle Astra et du conducteur . . . 26
- Vérifiez l’état du système . . . 27
- Configurer l’entrée pour l’équilibrage de charge . . . 32
- Connectez-vous à l’interface utilisateur du centre de contrôle Astra . . . 37
- Dépanner l’installation . . . 37
- Et la suite . . . 38
- Configurez un gestionnaire de certificats externe . . . 38

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.

Développez pour d'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**: Utiliser "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 àla 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 "restrictions 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.

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

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.

Téléchargez et extrayez Astra Control Center

1. Téléchargez le pack contenant Astra Control Center (`astra-control-center-[version].tar.gz`) du "Page de téléchargements d'Astra Control Center".
2. (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.

   Développez pour plus d'informations

   ```
   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.

3. 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 kubectl du plug-in NetApp Astra disponibles :

   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 fichier dont vous avez besoin pour votre système d'exploitation et votre architecture CPU 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 le `acc.manifest.bundle.yaml` et les répertoires suivants :

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

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=23.07.0-25
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/${
PACKAGEVERSION}/${astraImageNoPath}
done
```

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

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.07.0-25
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/23.07.0-25/image:version
```

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

1. Exportez la configuration kubeconfig pour le cluster hôte Astra Control Center :

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

> Avant de terminer l’installation, assurez-vous que votre kubeconfig pointe vers le cluster où vous souhaitez installer Astra Control Center.

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

Développez pour les étapes

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

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

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/23.07.0-25`).

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

> 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]
```

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]
```

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. Changer `ASTRA_IMAGE_REGISTRY` pour le `kube-rbac-proxy` image dans le chemin du registre où vous avez poussé les images dans un étape précédente.

c. Changer `ASTRA_IMAGE_REGISTRY` pour le `acc-operator-controller-manager` image dans le chemin du registre où vous avez poussé les images dans un étape précédente.

Développez pour l'exemple astra_control_Center_Operator_Deploy.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: ASTRA_IMAGE_REGISTRY/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_INSTALLTIMEOUT
          value: 5m
        image: ASTRA_IMAGE_REGISTRY/acc-operator:23.07.25
        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
```

Développer pour une réponse d'échantillon :

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

4. 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 `accountName` Chaîne du nom que vous souhaitez associer au compte Astra Control Center. Il ne peut y avoir qu'un seul nom de compte.	chaîne	`Example`

<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	`23.07.0-25`

<code>astraAddress</code>

Réglage	Guidage	Type	Exemple
`astraAddress`	Modifiez le `astraAddress` 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". REMARQUE : ne pas utiliser `http://` ou `https://` dans l'adresse. Copier ce FQDN pour l'utiliser dans un plus tard.	chaîne	`astra.example.com`

<code>autoSupport</code>

Vos sélections dans cette section déterminent si vous participerez à l'application de support proactif de NetApp, au conseiller numérique et à l'emplacement où les données sont envoyées. Une connexion Internet est requise (port 442) et toutes les données de support sont anonymisées.

Réglage	Utiliser	Guidage	Type	Exemple
`autoSupport.enrolled`	Soit `enrolled` ou `url` les champs doivent être sélectionnés	Changer `enrolled` Pour AutoSupport à `false` pour les sites sans connexion internet ou sans conservation `true` pour les sites connectés. Un réglage de `true` Permet d'envoyer des données anonymes à NetApp à des fins d'assistance. La sélection par défaut est `false` Aucune donnée de support n'est envoyée à NetApp.	Booléen	`false` (cette valeur est la valeur par défaut)
`autoSupport.url`	Soit `enrolled` ou `url` 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
`imageRegistry.name`	Obligatoire	Nom du registre d'images dans lequel vous avez poussé les images dans le étape précédente. Ne pas utiliser `http://` ou `https://` dans le nom du registre.	chaîne	`example.registry.com/astra`
`imageRegistry.secret`	Obligatoire si la chaîne que vous avez entrée pour `imageRegistry.name' requires a secret.` `IMPORTANT: If you are using a registry that does not require authorization, you must delete this` `secret` ligne comprise entre `imageRegistry` sinon, l'installation échouera.	Nom du secret Kubernetes utilisé pour s'authentifier auprès du registre d'images.	chaîne	`astra-registry-cred`

<code>storageClass</code>

Réglage	Guidage	Type	Exemple
`storageClass`	Modifiez le `storageClass` valeur à partir de `ontap-gold` À une autre ressource de classe de stockage Astra Trident, comme requis par votre installation. Lancer la commande `kubectl get sc` 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 (`astra-control-center-<version>.manifest`) Et sera utilisé pour ASTRA PVS. Si elle n'est pas définie, la classe de stockage par défaut sera utilisée. 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.	chaîne	`ontap-gold`

<code>volumeReclaimPolicy</code>

Réglage	Guidage	Type	Options
`volumeReclaimPolicy`	Cette règle définit la règle de récupération pour les volumes persistants d'Astra. Définition de cette règle sur `Retain` Conserve les volumes persistants après la suppression d'Astra. Définition de cette règle sur `Delete` supprime les volumes persistants après la suppression d'astra. Si cette valeur n'est pas définie, les PV sont conservés.	chaîne	• `Retain` (Il s'agit de la valeur par défaut) • `Delete`

<code>ingressType</code>

Réglage	Guidage	Type	Options
`ingressType`	Utilisez l'un des types d'entrées suivants : **`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	chaîne	• `Generic` (il s'agit de la valeur par défaut) • `AccTraefik`

<code>scaleSize</code>

Réglage	Guidage	Type	Options
`scaleSize`	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. CONSEIL : `Medium` 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.	chaîne	• `Small` • `Medium` (Il s'agit de la valeur par défaut)

<code>astraResourcesScaler</code>

Réglage	Guidage	Type	Options
`astraResourcesScaler`	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. Cependant, dans les scénarios utilisant des grappes de développement ou de test plus petites, le champ CR `astraResourcesScalar` peut être réglé sur `Off`. Cela désactive les demandes de ressources et permet un déploiement sur les clusters plus petits.	chaîne	• `Default` (Il s'agit de la valeur par défaut) • `Off`

<code>additionalValues</code>

Ajoutez les valeurs supplémentaires suivantes à l'Astra Control Center CR pour éviter un problème connu dans l'installation 23.07 :

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

- 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
`crds.externalCertManager`	Si vous utilisez un gestionnaire de certificats externe, modifiez-le `externalCertManager` à `true`. La valeur par défaut `false` Provoque l'installation d'Astra Control Center de ses propres CRD de cert Manager lors de l'installation. 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.	Booléen	`False` (cette valeur est la valeur par défaut)
`crds.externalTraefik`	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.	Booléen	`False` (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.

Développez pour l'exemple astra_control_Center.yaml

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

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]
```

> 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.

Développez pour obtenir une réponse d'échantillon

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

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

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

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

4. (En option) regarder le `acc-operator` journaux de suivi de la progression :

```
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  23.07.0-25
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 d'étapes pour certains types de contrôleurs d'entrée courants.

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. Rechercher 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.

Informations sur le copyright

Copyright © 2025 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 DÉLICTUELLE (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.