Commencez

Astra Control Center

NetApp
October 23, 2024

Sommaire

Commencez

Découvrez Astra Control

Astra Control est une solution de gestion du cycle de vie des données applicatives Kubernetes qui simplifie les opérations des applications avec état. Protégez, sauvegardez, répliquez et migrez facilement des workloads Kubernetes, et créez instantanément des clones d'applications de travail.

Caractéristiques

Astra Control offre des fonctionnalités stratégiques pour la gestion du cycle de vie des données d'application Kubernetes :

- Gérez automatiquement le stockage persistant
- Création de copies Snapshot et de sauvegardes à la demande intégrant la cohérence applicative
- Automatisation des opérations de sauvegarde et de snapshots basées sur des règles
- Migrez des applications et des données d'un cluster Kubernetes vers un autre
- Réplication d'applications sur un système distant à l'aide de la technologie NetApp SnapMirror (Astra Control Center)
- Clonage d'applications de la phase de transfert à la production
- Visualiser l'état de santé et de protection des applications
- Implémentation de vos workflows de sauvegarde et de migration à l'aide d'une interface utilisateur web ou d'une API

Modèles de déploiement

Astra Control est disponible dans deux modèles de déploiement :

- **Astra Control Service** : service géré par NetApp qui permet de gérer les données intégrant la cohérence applicative de clusters Kubernetes dans plusieurs environnements de fournisseurs cloud, ainsi que des clusters Kubernetes autogéré.
- **Astra Control Center** : logiciel autogéré qui assure une gestion des données compatible avec les applications de clusters Kubernetes exécutés dans votre environnement sur site. ASTRA Control Center peut également être installé sur des environnements de plusieurs fournisseurs cloud avec un système back-end de stockage NetApp Cloud Volumes ONTAP.

	Service Astra Control	**Centre de contrôle Astra**
Comment est-elle proposée ?	En tant que service cloud entièrement géré de NetApp	En tant que logiciel que vous pouvez télécharger, installer et gérer
Où est-il hébergé ?	Dans le cloud public de votre choix	Sur votre cluster Kubernetes
Comment est-elle mise à jour ?	Géré par NetApp	Vous gérez toutes les mises à jour

	Service Astra Control	Centre de contrôle Astra
Quels sont les systèmes back-end pris en charge ?	• Amazon Web Services : ◦ Amazon EBS ◦ Amazon FSX pour NetApp ONTAP ◦ "Cloud Volumes ONTAP" • Google Cloud : ◦ Disque persistant Google ◦ NetApp Cloud Volumes Service ◦ "Cloud Volumes ONTAP" • Microsoft Azure : ◦ Disques gérés Azure ◦ Azure NetApp Files ◦ "Cloud Volumes ONTAP" • Clusters autogérés : ◦ Amazon EBS ◦ Disque persistant Google ◦ Disques gérés Azure ◦ "Cloud Volumes ONTAP"	• Systèmes NetApp ONTAP AFF et FAS • "Cloud Volumes ONTAP"

Fonctionnement du service Astra Control

Astra Control Service est un service cloud géré par NetApp qui est constamment disponible et mis à jour avec les dernières fonctionnalités. Elle utilise plusieurs composants pour faciliter la gestion du cycle de vie des données des applications.

À un niveau élevé, le service de contrôle Astra fonctionne comme suit :

- Commencez avec le service Astra Control en configurant votre fournisseur de services cloud et en vous inscrivant à un compte Astra.
  - Pour les clusters GKE, Astra Control Service utilise "NetApp Cloud Volumes Service pour Google Cloud" Ou des disques persistants Google en tant que système de stockage back-end pour vos volumes persistants.
  - Pour les clusters AKS, Astra Control Service utilise "Azure NetApp Files" Ou des disques gérés Azure en tant que backend de stockage pour les volumes persistants.
  - Pour les clusters Amazon EKS, Astra Control Service utilise "Amazon Elastic Block Store" ou "Amazon FSX pour NetApp ONTAP" en tant que système back-end de stockage pour vos volumes persistants.
- Vous ajoutez votre première solution de calcul Kubernetes à Astra Control Service. Le service de contrôle d'Astra procède ensuite aux opérations suivantes :
  - Crée un magasin d'objets sur votre compte de fournisseur cloud, où sont stockées les copies de sauvegarde.

    Dans Azure, Astra Control Service crée également un groupe de ressources, un compte de stockage et des clés pour le conteneur Blob.

    - Crée un nouveau rôle d'administrateur et un compte de service Kubernetes sur le cluster.
    - Utilise ce nouveau rôle d'administrateur pour l'installation "Astra Trident" sur le cluster et pour créer une ou plusieurs classes de stockage.
    - Si vous utilisez une offre de stockage de service cloud NetApp comme système back-end de stockage, Astra Control Service utilise Astra Trident pour provisionner des volumes persistants pour vos applications. Si vous utilisez des disques gérés Amazon EBS ou Azure comme système de stockage principal, vous devez installer un pilote CSI spécifique au fournisseur. Les instructions d'installation sont fournies dans le "Configurer Amazon Web Services" et "Configuration de Microsoft Azure avec des disques gérés Azure".
- À ce stade, vous pouvez ajouter des applications à votre cluster. Les volumes persistants seront provisionnés sur la nouvelle classe de stockage par défaut.
- Utilisez ensuite le service Astra Control pour gérer ces applications, et commencez à créer des copies Snapshot, des sauvegardes et des clones.

Le plan gratuit d'Astra Control vous permet de gérer jusqu'à 10 espaces de noms dans votre compte. Si vous souhaitez gérer plus de 10 000 personnes, vous devrez configurer la facturation en passant du Plan gratuit au Plan Premium.

Fonctionnement du centre de contrôle Astra

Astra Control Center fonctionne localement dans votre propre cloud privé.

ASTRA Control Center prend en charge des clusters Kubernetes avec une classe de stockage basée sur Astra Trident avec un système back-end de stockage ONTAP 9.5 et versions ultérieures.

Dans un environnement connecté au cloud, Astra Control Center utilise Cloud Insights pour fournir des fonctionnalités avancées de surveillance et de télémétrie. En l'absence de connexion Cloud Insights, un monitoring et une télémétrie limités (7 jours de metrics) sont disponibles dans Astra Control Center, mais aussi exportés vers les outils de surveillance natifs de Kubernetes (comme Prometheus et Grafana) via des points de terminaison ouverts.

ASTRA Control Center est entièrement intégré à l'écosystème AutoSupport et Active IQ Digital Advisor (également appelé Digital Advisor) pour fournir aux utilisateurs et au support NetApp des informations sur le dépannage et l'utilisation.

Vous pouvez essayer Astra Control Center avec une licence d'évaluation intégrée de 90 jours. Pendant que vous évaluez Astra Control Center, vous pouvez obtenir de l'aide par e-mail et via des options communautaires. Vous avez également accès aux articles et à la documentation de la base de connaissances à partir du tableau de bord de support des produits.

Pour installer et utiliser Astra Control Center, vous devez vous en assurer "de formation".

À un niveau élevé, le centre de contrôle Astra ressemble à ce qui suit :

- Vous installez Astra Control Center dans votre environnement local. En savoir plus "Poser le centre de contrôle Astra".
- Vous avez effectué certaines tâches de configuration, telles que :
    - Configuration des licences.
    - Ajoutez votre premier cluster.
    - Ajout du stockage back-end découvert lorsque vous avez ajouté le cluster

- Ajoutez un compartiment de magasin d'objets pour stocker vos sauvegardes d'applications.

En savoir plus "Configurer le centre de contrôle Astra".

Vous pouvez ajouter des applications à votre cluster. Si certaines applications sont déjà gérées dans le cluster, vous pouvez aussi utiliser Astra Control Center pour les gérer. Utilisez ensuite Astra Control Center pour créer des copies Snapshot, des sauvegardes, des clones et des relations de réplication.

Pour en savoir plus

- "Documentation relative au service après-vente Astra Control"
- "Documentation Astra Control Center"
- "Documentation Astra Trident"
- "Utilisez l'API de contrôle Astra"
- "Documentation Cloud Insights"
- "Documentation ONTAP"

Exigences du centre de contrôle Astra

Commencez par vérifier que votre environnement opérationnel, vos clusters d'applications, vos applications, vos licences et votre navigateur Web sont prêts. Assurez-vous que votre environnement répond à ces exigences pour déployer et exploiter Astra Control Center.

- Environnements Kubernetes de cluster hôte pris en charge
- Ressources requises pour le cluster hôte
- Exigences d'Astra Trident
- Systèmes back-end
- Registre d'images
- Licence Astra Control Center
- Licences ONTAP
- Configuration réseau requise
- Entrée pour les clusters Kubernetes sur site
- Navigateurs Web pris en charge
- Exigences supplémentaires relatives aux clusters d'applications

Environnements Kubernetes de cluster hôte pris en charge

ASTRA Control Center a été validé avec les environnements hôtes Kubernetes suivants :

> Assurez-vous que l'environnement Kubernetes que vous choisissez d'héberger Astra Control Center répond aux exigences de ressources de base indiquées dans la documentation officielle de l'environnement.

Distribution Kubernetes sur le cluster hôte	Versions prises en charge
Azure Kubernetes Service sur Azure Stack HCI	Pile Azure HCI 21H2 et 22H2 avec AKS 1.24.x et 1.25.x
Anthos de Google	1.14 à 1.16 (voir Exigences d'entrée de Google Anthos)
Kubernetes (en amont)	1.25 à 1.27 (Astra Trident 22.10 ou version ultérieure requise pour Kubernetes 1.25 ou version ultérieure)
Rancher Kubernetes Engine (RKE)	RKE 1.3 avec Rancher Manager 2.6 RKE 1.4 avec Rancher Manager 2.7 RKE 2 (v1.24.x) avec Rancher 2.6 RKE 2 (v1.25.x) avec Rancher 2.7
Plateforme de conteneurs Red Hat OpenShift	4.11 à 4.13

Ressources requises pour le cluster hôte

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

> 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.

- **Extensions CPU** : les CPU de tous les noeuds de l'environnement d'hébergement doivent avoir des extensions AVX activées.
- **Nœuds de travail** : au moins 3 nœuds de travail au total, avec 4 cœurs de processeur et 12 Go de RAM chacun

Exigences d'Astra Trident

Assurez-vous de répondre aux exigences Astra Trident suivantes, spécifiques aux besoins de votre environnement :

- **Version minimale pour une utilisation avec Astra Control Center** : Astra Trident 22.10 ou version ultérieure installée et configurée.
- **Réplication SnapMirror** : Astra Trident 22.10 ou version ultérieure installée pour la réplication d'applications basée sur SnapMirror.
- **Pour la prise en charge de Kubernetes 1.25 ou version ultérieure** : Astra Trident 22.10 ou version ultérieure installée pour les clusters Kubernetes 1.25 ou version ultérieure (vous devez effectuer une mise à niveau vers Astra Trident 22.10 avant de procéder à la mise à niveau vers Kubernetes 1.25 ou version ultérieure)
- **Configuration ONTAP avec Astra Trident** :
  - **Classe de stockage** : configurez au moins une classe de stockage Astra Trident sur le cluster. Si une classe de stockage par défaut est configurée, assurez-vous qu'elle est la seule classe de stockage avec la désignation par défaut.
  - **Pilotes de stockage et nœuds worker** : Assurez-vous que les nœuds worker de votre cluster sont configurés avec les pilotes de stockage appropriés afin que les pods puissent interagir avec le stockage back-end. Le centre de contrôle Astra prend en charge les pilotes ONTAP suivants fournis

par Astra Trident :

- `ontap-nas`
- `ontap-san`
- `ontap-san-economy` (la réplication d'application n'est pas disponible avec ce type de classe de stockage)
- `ontap-nas-economy` (les snapshots, les règles de réplication et les règles de protection ne sont pas disponibles avec ce type de classe de stockage)

Systèmes back-end

Assurez-vous que vous disposez d'un back-end pris en charge avec une capacité suffisante.

- **Capacité de stockage requise** : au moins 500 Go disponibles
- **Systèmes back-end pris en charge** : Astra Control Center prend en charge les systèmes back-end de stockage suivants :
  - NetApp ONTAP 9.8 ou version ultérieure des systèmes AFF, FAS et ASA
  - NetApp ONTAP Select 9.8 ou version ultérieure
  - NetApp Cloud Volumes ONTAP 9.8 ou version ultérieure
  - Longhorn 1.5.0 ou version ultérieure
    - Nécessite la création manuelle d'un objet VolumeSnapshotClass. Reportez-vous à la "Documentation Longhorn" pour obtenir des instructions.
  - NetApp MetroCluster
    - Les clusters Kubernetes gérés doivent se trouver dans une configuration étendue.

Licences ONTAP

Pour utiliser Astra Control Center, vérifiez que vous disposez des licences ONTAP suivantes, en fonction de ce que vous devez accomplir :

- FlexClone
- SnapMirror : en option. Elle est nécessaire uniquement pour la réplication vers des systèmes distants à l'aide de la technologie SnapMirror. Reportez-vous à la section "Informations sur la licence SnapMirror".
- Licence S3 : en option. Nécessaire uniquement pour les compartiments ONTAP S3

Pour vérifier si votre système ONTAP dispose des licences requises, reportez-vous à la section "Gérer les licences ONTAP".

NetApp MetroCluster

Lorsque vous utilisez NetApp MetroCluster en tant que système back-end de stockage, vous devez spécifier une LIF de gestion de SVM en tant qu'option back-end dans le pilote Astra Trident que vous utilisez.

Pour configurer le LIF MetroCluster, consultez la documentation d'Astra Trident pour plus d'informations sur chaque pilote :

- "SAN"
- "NAS"

Registre d'images

Vous devez disposer d'un registre d'images Docker privé sur lequel vous pouvez envoyer les images de build d'Astra Control Center. Vous devez fournir l'URL du registre d'images où vous allez télécharger les images.

Licence Astra Control Center

ASTRA Control Center requiert une licence Astra Control Center. Lorsque vous installez Astra Control Center, une licence d'évaluation intégrée de 90 jours pour 4,800 UC est déjà activée. Si vous avez besoin de davantage de capacité ou de conditions d'évaluation différentes, ou si vous souhaitez effectuer une mise à niveau vers une licence complète, vous pouvez obtenir une autre licence d'évaluation ou une licence complète auprès de NetApp. Vous devez disposer d'une licence pour protéger vos applications et vos données.

Vous pouvez essayer Astra Control Center en vous inscrivant pour un essai gratuit. Vous pouvez vous inscrire en vous inscrivant "ici".

Pour configurer la licence, reportez-vous à la section "utilisez une licence d'essai gratuite de 90 jours".

Pour en savoir plus sur le fonctionnement des licences, reportez-vous à la section "Licences".

Configuration réseau requise

Configurez votre environnement opérationnel pour vous assurer qu'Astra Control Center peut communiquer correctement. Les configurations réseau suivantes sont requises :

- **Adresse FQDN** : vous devez avoir une adresse FQDN pour Astra Control Center.
- **Accès à Internet** : vous devez déterminer si vous avez un accès extérieur à Internet. Si ce n'est pas le cas, certaines fonctionnalités peuvent être limitées, comme la réception de données de surveillance et de metrics depuis NetApp Cloud Insights ou l'envoi de packs de support au "Site de support NetApp".
- **Port Access** : l'environnement opérationnel qui héberge Astra Control Center communique avec les ports TCP suivants. Veillez à ce que ces ports soient autorisés par le biais de pare-feu et configurez des pare-feu pour autoriser tout trafic de sortie HTTPS provenant du réseau Astra. Certains ports nécessitent une connectivité entre l'environnement hébergeant le centre de contrôle Astra et chaque cluster géré (le cas échéant).

> Vous pouvez déployer Astra Control Center dans un cluster Kubernetes à double pile, et Astra Control Center peut gérer les applications et les systèmes back-end de stockage qui ont été configurés pour un fonctionnement à double pile. Pour plus d'informations sur la configuration requise pour les clusters à double pile, consultez le "Documentation Kubernetes".

Source	Destination	Port	Protocole	Objectif
PC client	Centre de contrôle Astra	443	HTTPS	Accès à l'interface utilisateur/à l'API : assurez-vous que ce port est ouvert à la fois entre le cluster hébergeant Astra Control Center et chaque cluster géré

Source	Destination	Port	Protocole	Objectif
Consommateurs de metrics	Nœud de travail Astra Control Center	9090	HTTPS	Communication de données de metrics : assurez-vous que chaque cluster géré peut accéder à ce port sur le cluster hébergeant Astra Control Center (communication bidirectionnelle requise).
Centre de contrôle Astra	Service Cloud Insights hébergé (https://www.netapp.com/cloud-services/cloud-insights/)	443	HTTPS	Communication avec Cloud Insights
Centre de contrôle Astra	Fournisseur de compartiments de stockage Amazon S3	443	HTTPS	Communications de stockage Amazon S3
Centre de contrôle Astra	NetApp AutoSupport (https://support.netapp.com)	443	HTTPS	Communication avec NetApp AutoSupport

Entrée pour les clusters Kubernetes sur site

Vous pouvez choisir le type d'entrée de réseau utilisé par le centre de contrôle Astra. Par défaut, Astra Control Center déploie la passerelle Astra Control Center (service/trafik) comme ressource à l'échelle du cluster. Astra Control Center prend également en charge l'utilisation d'un équilibreur de charge de service, s'ils sont autorisés dans votre environnement. Si vous préférez utiliser un équilibreur de charge de service et que vous n'avez pas encore configuré, vous pouvez utiliser l'équilibreur de charge MetalLB pour attribuer automatiquement 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.

> L'équilibreur de charge doit utiliser une adresse IP située dans le même sous-réseau que les adresses IP du nœud de travail de l'Astra Control Center.

Pour plus d'informations, reportez-vous à la section "Configurer l'entrée pour l'équilibrage de charge".

Exigences d'entrée de Google Anthos

Lorsque vous hébergez Astra Control Center sur un cluster Google Anthos, notez que Google Anthos inclut par défaut l'équilibreur de charge MetalLB et le service d'entrée Istio, ce qui vous permet d'utiliser simplement les fonctionnalités d'entrée génériques d'Astra Control Center lors de l'installation. Reportez-vous à la section "Configurer le centre de contrôle Astra" pour plus d'informations.

Navigateurs Web pris en charge

Astra Control Center prend en charge les versions récentes de Firefox, Safari et Chrome avec une résolution

minimale de 1280 x 720.

Exigences supplémentaires relatives aux clusters d'applications

Gardez à l'esprit ces exigences si vous prévoyez d'utiliser ces caractéristiques du centre de contrôle Astra :

- **Configuration requise pour le cluster d'applications** : "Exigences de gestion du cluster"
  - * Exigences des applications gérées* : "De gestion des applications"
  - **Exigences supplémentaires pour la réplication d'applications** : "Conditions préalables à la réplication"

Et la suite

Afficher le "démarrage rapide" présentation.

Démarrage rapide pour Astra Control Center

Voici un aperçu des étapes à suivre pour commencer à utiliser le centre de contrôle Astra. Les liens de chaque étape vous mènent à une page qui fournit plus de détails.

1 Vérifiez la configuration des clusters Kubernetes

Assurez-vous que votre environnement répond aux exigences suivantes :

Cluster Kubernetes

- "Assurez-vous que votre cluster hôte répond aux exigences de l'environnement opérationnel"
- "Configuration de la détection d'entrée pour l'équilibrage de la charge sur les clusters Kubernetes sur site"

Intégration du stockage

- "Vérifiez que votre environnement inclut la version prise en charge d'Astra Trident"
- "Préparez les nœuds worker"
- "Configurer le système back-end de stockage Astra Trident"
- "Configurez des classes de stockage Astra Trident"
- "Installation du contrôleur de snapshot de volume Astra Trident"
- "Créer une classe de snapshot de volume"

Informations d'identification ONTAP

- "Configurez les identifiants ONTAP"

2 Téléchargez et installez Astra Control Center

Effectuez les tâches d'installation suivantes :

- "Téléchargez Astra Control Center à partir de la page de téléchargements du site de support NetApp"

- Obtenez le fichier de licence NetApp :
  - Si vous évaluez Astra Control Center, une licence d'évaluation intégrée est déjà incluse
  - "Si vous avez déjà acheté Astra Control Center, générez votre fichier de licence"
- "Poser le centre de contrôle Astra"
- "Effectuez d'autres étapes de configuration facultatives"

3 Effectuez certaines tâches de configuration initiales

Effectuez quelques tâches de base pour commencer :

- "Ajouter une licence"
- "Préparez votre environnement à la gestion du cluster"
- "Ajouter un cluster"
- "Ajout d'un système back-end"
- "Ajouter un godet"

4 Utilisez Astra Control Center

Une fois la configuration d'Astra Control Center terminée, utilisez l'interface utilisateur d'Astra Control ou le "API de contrôle Astra" pour commencer à gérer et à protéger les applications :

- "Gérer des applications": Définissez les ressources à gérer.
- "Protégez vos applications": Configurer des stratégies de protection et répliquer, cloner et migrer des applications.
- "Gestion des comptes": Utilisateurs, rôles, LDAP, informations d'identification, etc.
- "Vous pouvez également vous connecter à Cloud Insights": Permet d'afficher des mesures sur l'état de santé de votre système.

Pour en savoir plus

- "Utilisez l'API de contrôle Astra"
- "Mettez à niveau Astra Control Center"
- "Aidez-vous d'Astra Control"

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.

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 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 "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.

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. Utiliser 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. 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.

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 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
```

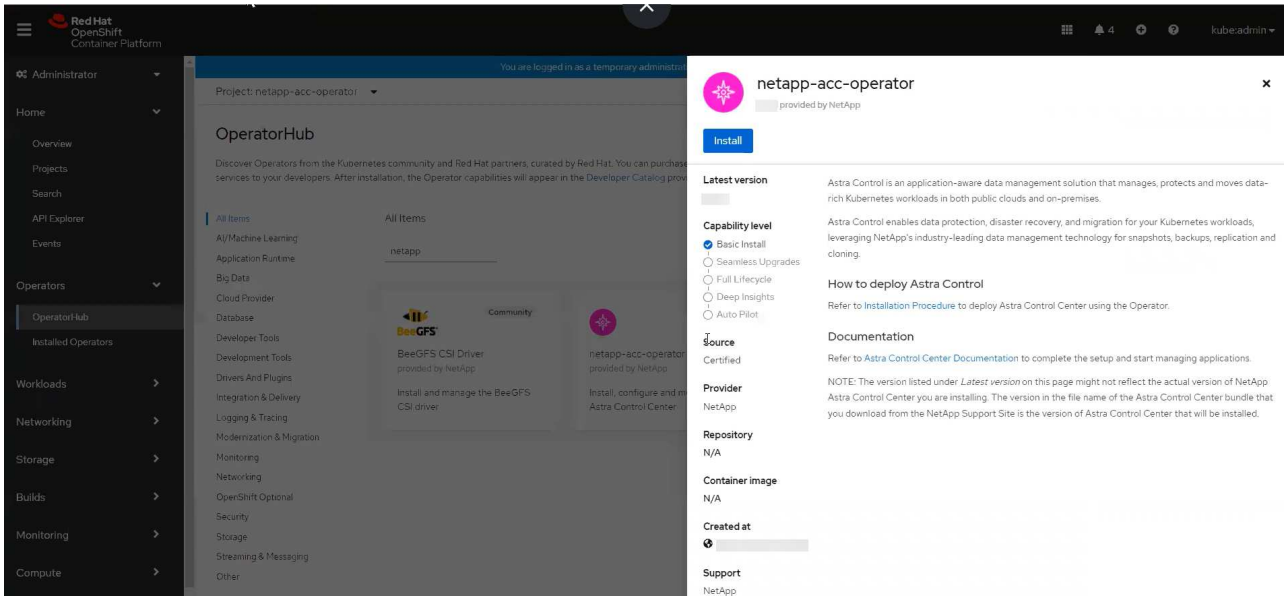
Recherchez la page d'installation de l'opérateur

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

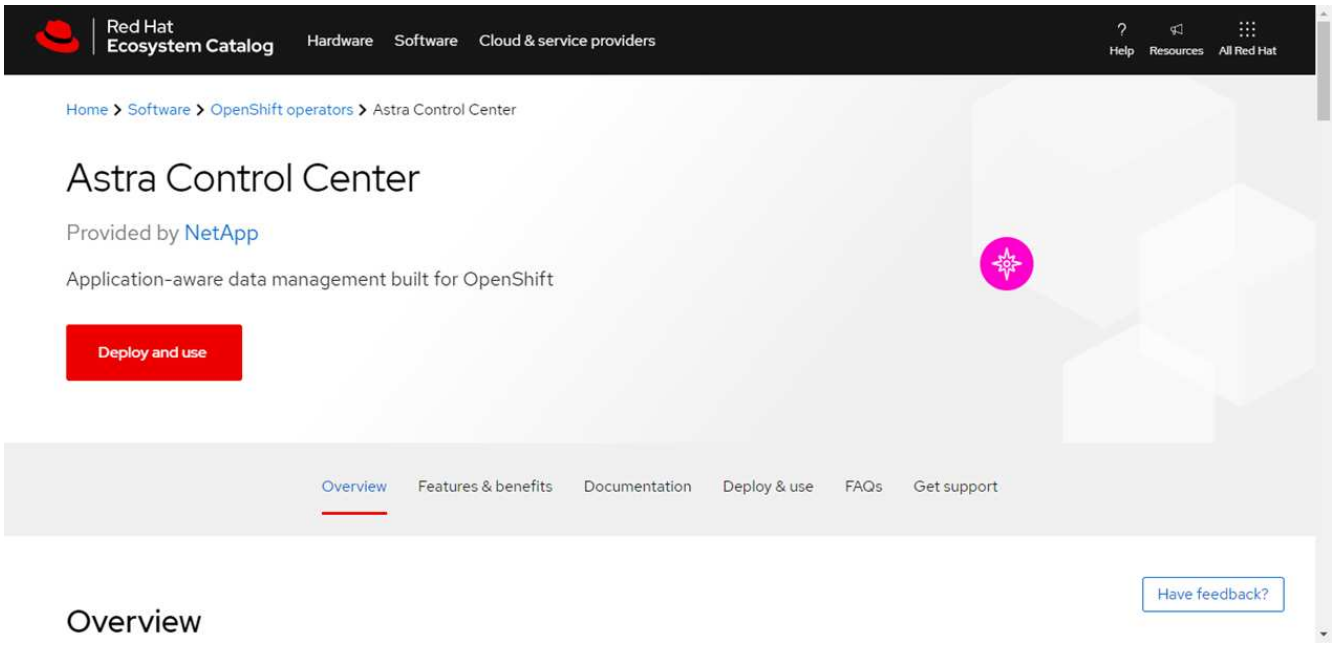
- Depuis la console Web Red Hat OpenShift :
  i. Connectez-vous à l'interface utilisateur de OpenShift Container Platform.
  ii. Dans le menu latéral, sélectionnez **Operators > OperatorHub**.

> Vous ne pouvez effectuer la mise à niveau que vers la version actuelle d'Astra Control Center à l'aide de cet opérateur.

  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 **déployer 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.

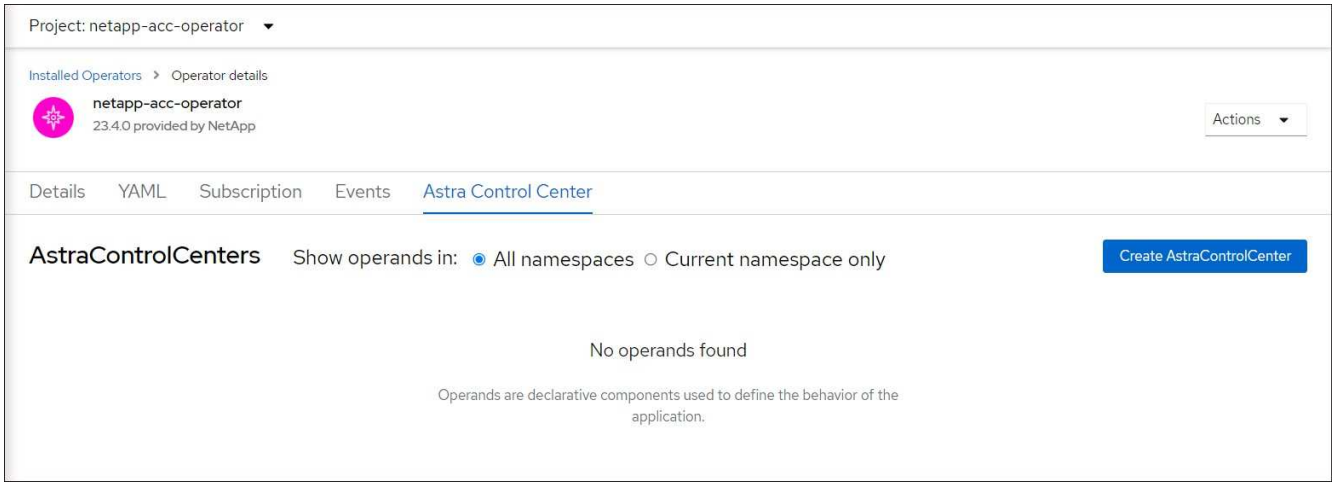
   c. Sélectionnez **installer**.

   > 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 23.07.0-25.

   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 4.11 à 4.13

> 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
Astra Trident (installé dans le cadre de la découverte du cluster Kubernetes dans NetApp BlueXP, anciennement Cloud Manager)	ASTRA Trident 22.10 ou version ultérieure est installé et configuré et NetApp ONTAP version 9.8 ou version ultérieure en tant que système back-end de stockage

Composant	Conditions requises
Registre d'images	NetApp fournit un registre que vous pouvez utiliser pour obtenir les images de build d'Astra Control Center : `http://netappdownloads.jfrog.io/docker-astra-control-prod` Contactez le support NetApp pour obtenir des instructions sur l'utilisation de ce registre d'images pendant le processus d'installation d'Astra Control Center. Si vous ne parvenez pas à accéder au registre d'images NetApp, vous devez disposer d'un registre privé existant, tel qu'AWS Elastic Container Registry (ECR), auquel vous pouvez envoyer les images de build d'Astra Control Center. Vous devez fournir l'URL du registre d'images où vous allez télécharger les images. 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.
Configuration d'Astra Trident et ONTAP	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 : • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

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, configurer des instances de calcul EC2 et créer un compartiment AWS S3. Si vous ne pouvez pas accéder au Registre d'images NetApp Astra Control Center, Vous devrez également créer un registre de conteneurs élastiques (ECR) pour héberger les images d'Astra Control Center et les transmettre à ce registre.

Suivez la documentation AWS pour suivre la procédure ci-dessous. Voir "Documentation d'installation d'AWS".

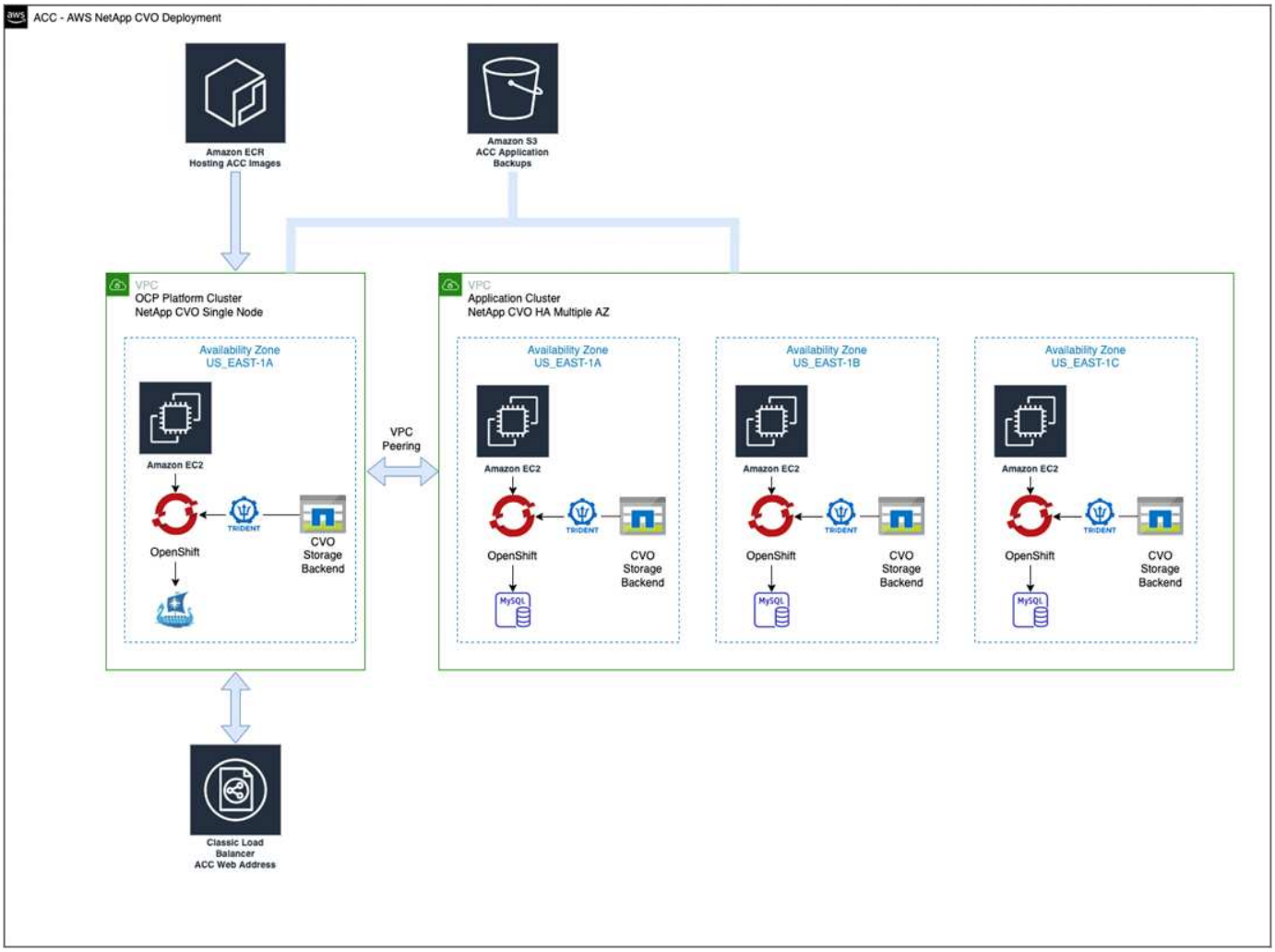
1. Créez un réseau virtuel AWS.
2. Vérifiez les instances de calcul EC2. Il peut s'agir d'un serveur bare Metal ou de machines virtuelles dans AWS.
3. Si le type d'instance ne correspond pas déjà aux exigences de ressources minimales Astra pour les nœuds maîtres et workers, modifiez le type d'instance 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. (Facultatif) si vous ne pouvez pas accéder au Registre d'images NetApp, procédez comme suit :
   a. Créez un registre AWS Elastic Container Registry (ECR) pour héberger toutes les images d'Astra Control Center.

      > 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.

   b. Envoyez les images d'Astra Control Center vers votre 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) 4.11 à 4.13
- 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 22.10 ou version ultérieure est installé et configuré et NetApp ONTAP version 9.8 ou version ultérieure en tant que système back-end de stockage
Registre d'images	NetApp fournit un registre que vous pouvez utiliser pour obtenir les images de build d'Astra Control Center : `http://netappdownloads.jfrog.io/docker-astra-control-prod` Contactez le support NetApp pour obtenir des instructions sur l'utilisation de ce registre d'images pendant le processus d'installation d'Astra Control Center. Si vous ne parvenez pas à accéder au Registre d'images NetApp, vous devez disposer d'un registre privé existant, tel que le Registre de conteneurs Google, auquel vous pouvez envoyer les images de build d'Astra Control Center. Vous devez fournir l'URL du registre d'images où vous allez télécharger les images. Vous devez activer l'accès anonyme pour extraire les images Restic pour les sauvegardes.

Composant	Conditions requises
Configuration d'Astra Trident et ONTAP	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 : • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

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, configurer des instances de calcul et créer un stockage objet Google Cloud. Si vous ne pouvez pas accéder au Registre d'images NetApp Astra Control Center, Vous devrez également créer un registre de conteneurs Google pour héberger les images d'Astra Control Center et les envoyer dans 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. (Facultatif) si vous ne pouvez pas accéder au Registre d'images NetApp, procédez comme suit :
   a. Créez un registre de conteneurs Google pour héberger les images d'Astra Control Center.
   b. 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 d'Astra Control Center peuvent être transmises à ce registre en saisissant 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 registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image
    docker push $GCP_CR_REGISTRY/$image
done < astra-control-center-22.04.41.manifest
```

1. 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) 4.11 à 4.13
- 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 22.10 ou version ultérieure installée et configurée et NetApp ONTAP version 9.8 ou ultérieure seront utilisés en tant que système back-end de stockage
Registre d'images	NetApp fournit un registre que vous pouvez utiliser pour obtenir les images de build d'Astra Control Center : `http://netappdownloads.jfrog.io/docker-astra-control-prod` Contactez le support NetApp pour obtenir des instructions sur l'utilisation de ce registre d'images pendant le processus d'installation d'Astra Control Center. Si vous ne parvenez pas à accéder au registre d'images NetApp, vous devez disposer d'un registre privé existant, tel qu'Azure Container Registry (ACR), auquel vous pouvez envoyer les images de build d'Astra Control Center. Vous devez fournir l'URL du registre d'images où vous allez télécharger les images. Vous devez activer l'accès anonyme pour extraire les images Restic pour les sauvegardes.
Configuration d'Astra Trident et ONTAP	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 : • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

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, configurer des instances de calcul et créer un conteneur Azure Blob. Si vous ne pouvez pas accéder au Registre d'images NetApp Astra Control Center, Vous devrez également créer un Registre de conteneurs Azure (ACR) pour héberger les images d'Astra Control Center et envoyer les images vers 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. (Facultatif) si vous ne pouvez pas accéder au Registre d'images NetApp, procédez comme suit :

   a. Créez un registre de conteneurs Azure (ACR) pour héberger les images d'Astra Control Center.
   b. Configurez l'accès ACR pour Docker Push/Pull pour toutes les images d'Astra Control Center.
   c. Envoyez les images d'Astra Control Center vers ce registre à l'aide du script suivant :

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

      Exemple :

```
manifestfile=astra-control-center-<version>.manifest
AZ_ACR_REGISTRY=<target image registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRYY/$image
    docker push $AZ_ACR_REGISTRY/$image
done < astra-control-center-22.04.41.manifest
```

8. 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".

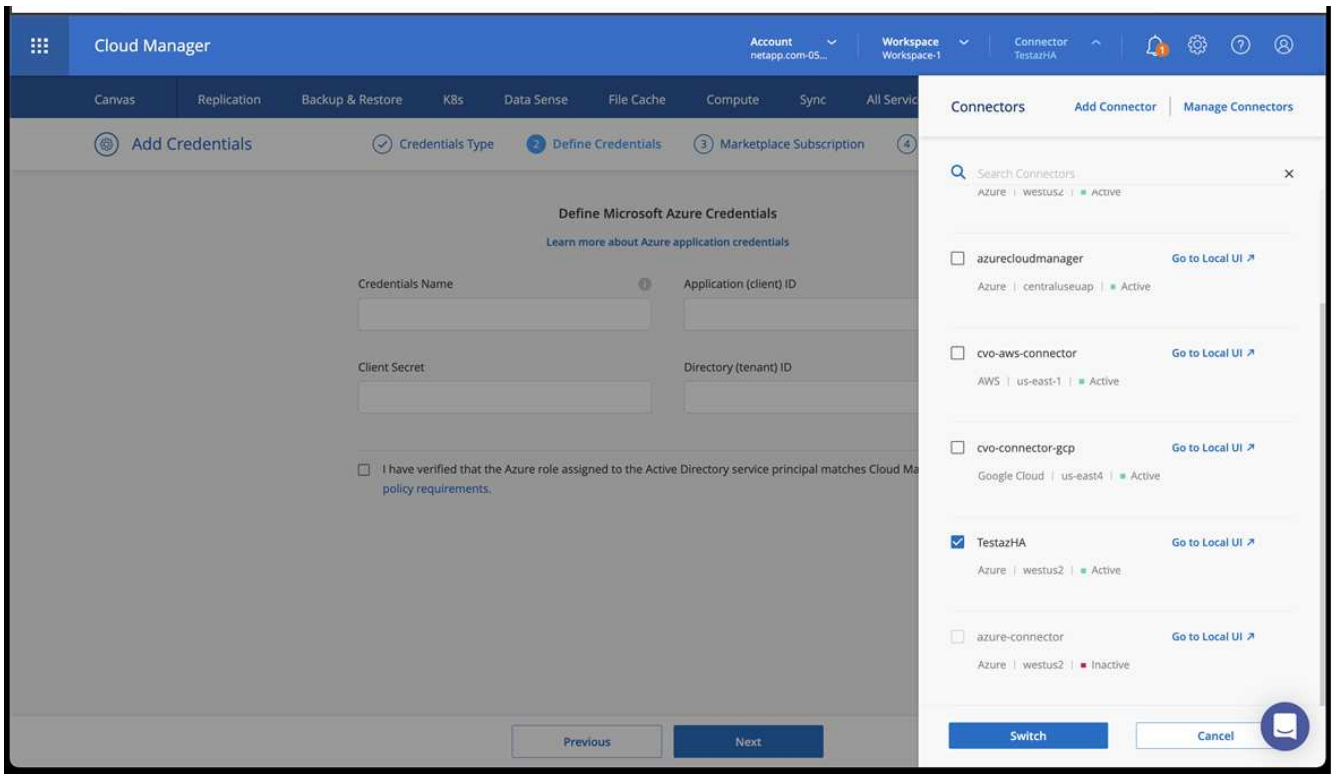
Avant de commencer

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

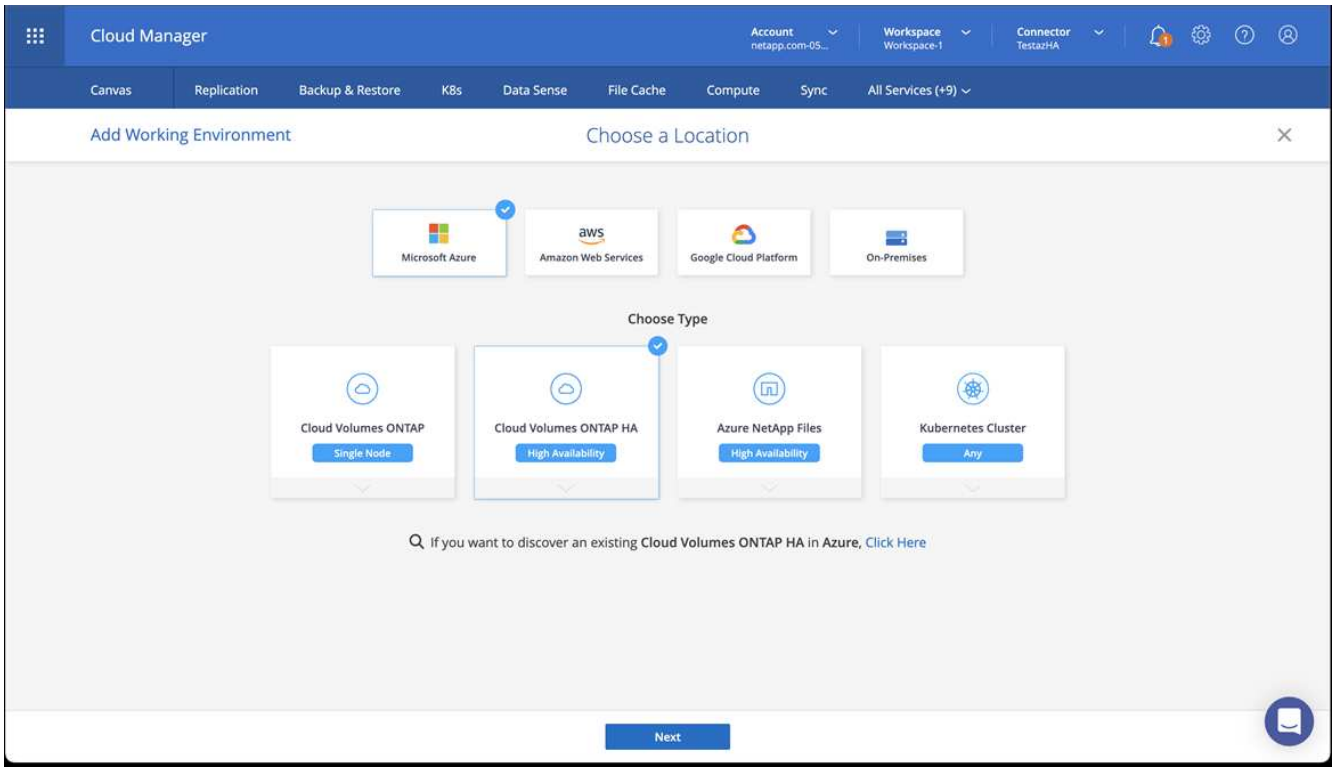
Étapes

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

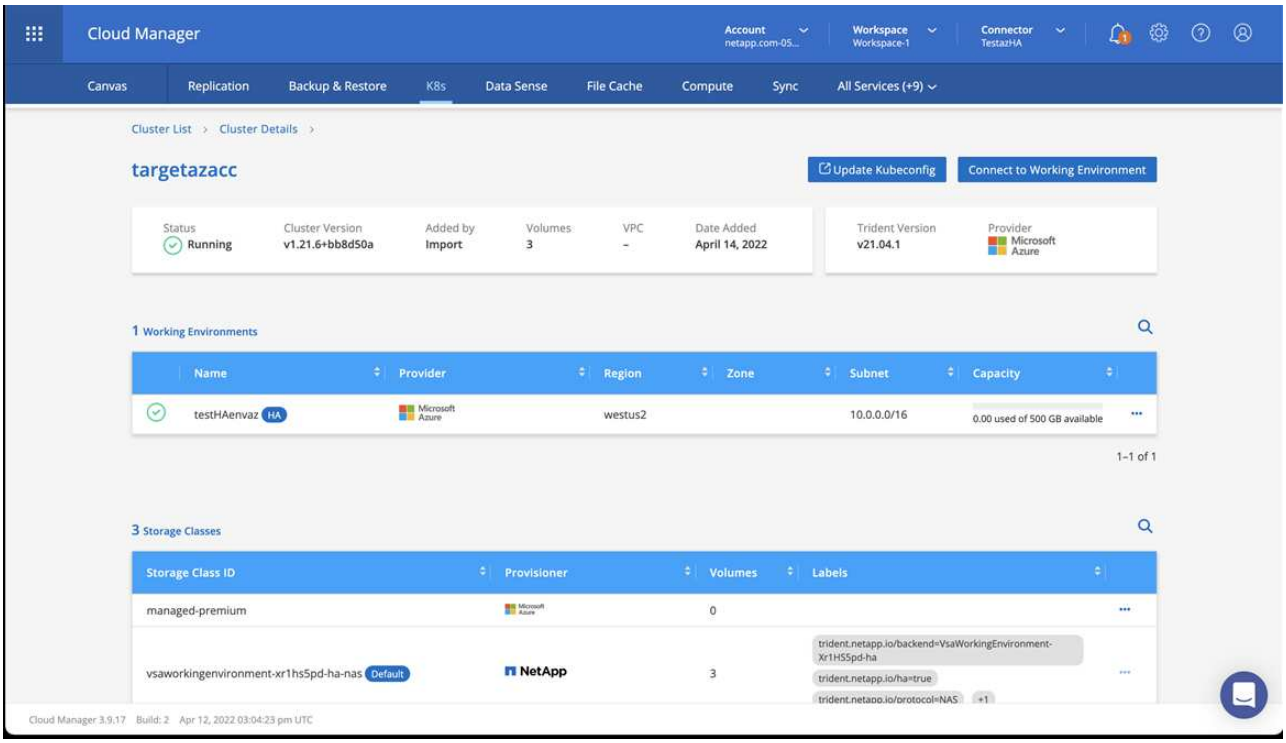


4. Créez un environnement de travail pour votre environnement cloud.
   a. Emplacement : « Microsoft Azure ».
   b. Type : « Cloud Volumes ONTAP HA ».



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]
```

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>
```

   CRD :

```
#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>
```

Réponse :

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

5. 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>
```

Réponse :

```
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 le TLS stocke CRD pour pointer vers votre nouveau nom de secret de certificat à l'aide de la commande et de l'exemple suivants, en remplaçant les valeurs d'espace réservé entre parenthèses <> par les informations appropriées

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

CRD :

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

8. 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.io -n <ACC-deployment-namespace>
```

CRD :

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

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

Configurer le centre de contrôle Astra

Une fois Astra Control Center installé, connectez-vous à l'interface utilisateur et modifiez votre mot de passe, vous pouvez configurer une licence, ajouter des clusters, activer l'authentification, gérer le stockage et ajouter des compartiments.

Tâches

- Ajoutez une licence pour Astra Control Center
- Préparez votre environnement à la gestion des clusters avec Astra Control
- Ajouter un cluster
- Activez l'authentification sur le back-end de stockage ONTAP
- Ajout d'un système back-end
- Ajouter un godet

Ajoutez une licence pour Astra Control Center

Lorsque vous installez Astra Control Center, une licence d'évaluation intégrée est déjà installée. Si vous évaluez Astra Control Center, vous pouvez ignorer cette étape.

Vous pouvez ajouter une nouvelle licence à l'aide de l'interface utilisateur Astra Control ou "API de contrôle Astra".

Les licences Astra Control Center mesurent les ressources CPU avec des unités de processeur Kubernetes et représentent les ressources de processeur attribuées aux nœuds de travail de tous les clusters Kubernetes gérés. Les licences dépendent de l'utilisation des processeurs virtuels. Pour plus d'informations sur le calcul des licences, reportez-vous à la section "Licences".

> Si votre installation dépasse le nombre de processeurs sous licence, Astra Control Center vous empêche de gérer de nouvelles applications. Une alerte s'affiche lorsque la capacité est dépassée.

> Pour mettre à jour une évaluation existante ou une licence complète, reportez-vous à la section "Mettre à jour une licence existante".

Avant de commencer

- Accès à une instance Astra Control Center récemment installée.
- Autorisations de rôle d'administrateur.

- A "Fichier de licence NetApp" (NLF).

Étapes

1. Connectez-vous à l'interface utilisateur du centre de contrôle Astra.
2. Sélectionnez **compte** > **Licence**.
3. Sélectionnez **Ajouter licence**.
4. Accédez au fichier de licence (NLF) que vous avez téléchargé.
5. Sélectionnez **Ajouter licence**.

La page **Account** > **License** affiche les informations de licence, la date d'expiration, le numéro de série de licence, l'ID de compte et les unités UC utilisées.

> Si vous disposez d'une licence d'évaluation et que vous n'envoyez pas de données à AutoSupport, assurez-vous de stocker votre identifiant de compte pour éviter toute perte de données en cas de défaillance d'Astra Control Center.

Préparez votre environnement à la gestion des clusters avec Astra Control

Avant d'ajouter un cluster, assurez-vous que les conditions préalables suivantes sont remplies. Vous devez également effectuer des vérifications d'admissibilité pour vous assurer que votre cluster est prêt à être ajouté au Centre de contrôle Astra et créer des rôles pour la gestion du cluster.

Avant de commencer

- Assurez-vous que les nœuds workers de votre cluster sont configurés avec les pilotes de stockage appropriés afin que les pods puissent interagir avec le système de stockage back-end.
- Votre environnement répond au "de l'environnement opérationnel" Pour Astra Trident et Astra Control Center.
- Si vous ajoutez le cluster à l'aide d'un fichier kubeconfig qui fait référence à une autorité de certification privée (CA), ajoutez la ligne suivante au `cluster` section du fichier kubeconfig. Cela permet à Astra Control d'ajouter le cluster :

```
insecure-skip-tls-verify: true
```

- Une version d'Astra Trident "Pris en charge par Astra Control Center" est installé :

  > C'est possible "Déployez Astra Trident" À l'aide de l'opérateur Astra Trident (manuellement ou à l'aide du tableau Helm) ou `tridentctl`. Avant d'installer ou de mettre à niveau Astra Trident, consultez le "systèmes front-end, systèmes back-end et configurations hôte pris en charge".

  - **Système back-end de stockage Astra Trident configuré** : au moins un système back-end de stockage Astra Trident doit l'être "configuré" sur le cluster.
  - **Classes de stockage Astra Trident configurées** : au moins une classe de stockage Astra Trident doit être "configuré" sur le cluster. 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.
  - **Contrôleur de snapshot de volume Astra Trident et classe de snapshot de volume installés et configurés** : le contrôleur de snapshot de volume doit être "installé" Il est ainsi possible de créer des

snapshots dans Astra Control. Au moins un Astra Trident `VolumeSnapshotClass` a été "configuration" par un administrateur.

- **Kubeconfig accessible**: Vous avez accès au "configuration par défaut du cluster" ça "vous avez configuré lors de l'installation".
- **Informations d'identification ONTAP** : vous avez besoin d'informations d'identification ONTAP et d'un superutilisateur et d'un ID utilisateur définis sur le système ONTAP de sauvegarde pour sauvegarder et restaurer des applications avec le Centre de contrôle Astra.

  Exécutez les commandes suivantes dans la ligne de commande ONTAP :

  ```
  export-policy rule modify -vserver <storage virtual machine name>
  -policyname <policy name> -ruleindex 1 -superuser sys
  export-policy rule modify -vserver <storage virtual machine name>
  -policyname <policy name> -ruleindex 1 -anon 65534
  ```

- **Rancher uniquement**: Lorsque vous gérez des clusters d'applications dans un environnement Rancher, modifiez le contexte par défaut du cluster d'applications dans le fichier kubeconfig fourni par Rancher pour utiliser un contexte de plan de contrôle au lieu du contexte du serveur d'API Rancher. La charge est réduite sur le serveur API Rancher et les performances sont améliorées.

Effectuer des vérifications d'éligibilité

Effectuez les contrôles d'éligibilité suivants pour vous assurer que votre grappe est prête à être ajoutée au Centre de contrôle Astra.

Étapes

1. Vérifiez la version d'Astra Trident.

   ```
   kubectl get tridentversions -n trident
   ```

   Si Astra Trident existe, le résultat de cette commande est similaire à ce qui suit :

   ```
   NAME      VERSION
   trident   23.XX.X
   ```

   Si Astra Trident n'existe pas, le résultat est similaire à ce qui suit :

   ```
   error: the server doesn't have a resource type "tridentversions"
   ```

   > Si Astra Trident n'est pas installé ou si la version installée n'est pas la plus récente, vous devez installer la dernière version d'Astra Trident avant de continuer. Reportez-vous à la "Documentation Astra Trident" pour obtenir des instructions.

2. Assurez-vous que les pods fonctionnent :

```
kubectl get pods -n trident
```

3. Déterminez si les classes de stockage utilisent les pilotes Astra Trident pris en charge. Le nom de provisionnement doit être `csi.trident.netapp.io`. Voir l'exemple suivant :

```
kubectl get sc
```

Exemple de réponse :

```
NAME                   PROVISIONER            RECLAIMPOLICY
VOLUMEBINDINGMODE  ALLOWVOLUMEEXPANSION  AGE
ontap-gold (default)  csi.trident.netapp.io  Delete         Immediate
true                   5d23h
```

Créez un kubeconfig pour le rôle de cluster

Vous pouvez éventuellement créer une autorisation limitée ou un rôle d'administrateur d'autorisations étendues pour Astra Control Center. Il ne s'agit pas d'une procédure requise pour la configuration d'Astra Control Center, car vous avez déjà configuré un kubeconfig dans le cadre du "processus d'installation".

Cette procédure vous aide à créer un kubeconfig distinct si l'un des scénarios suivants s'applique à votre environnement :

- Vous souhaitez limiter les autorisations Astra Control sur les clusters qu'il gère
- Vous utilisez plusieurs contextes et ne pouvez pas utiliser le kubeconfig Astra Control par défaut configuré lors de l'installation, sinon un rôle limité avec un seul contexte ne fonctionnera pas dans votre environnement

Avant de commencer

Assurez-vous que vous disposez des éléments suivants pour le cluster que vous souhaitez gérer avant d'effectuer la procédure suivante :

- kubectl v1.23 ou version ultérieure installée
- Accès kubectl au cluster que vous souhaitez ajouter et gérer avec Astra Control Center

> Pour cette procédure, il n'est pas nécessaire d'avoir un accès kubectl au cluster qui exécute Astra Control Center.

- Un kubeconfig actif pour le cluster que vous avez l'intention de gérer avec des droits d'administrateur de cluster pour le contexte actif

Étapes

1. Créer un compte de service :

   a. Créez un fichier de compte de service appelé `astracontrol-service-account.yaml`.

Ajustez le nom et l'espace de noms selon vos besoins. Si des modifications sont apportées ici, vous devez appliquer les mêmes modifications dans les étapes suivantes.

```
<strong>astracontrol-service-account.yaml</strong>
```

+

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astracontrol-service-account
  namespace: default
```

a. Appliquer le compte de service :

```
kubectl apply -f astracontrol-service-account.yaml
```

2. Créez l'un des rôles de cluster suivants avec des autorisations suffisantes pour qu'un cluster soit géré par Astra Control :

   - **Rôle de cluster limité** : ce rôle contient les autorisations minimales nécessaires à la gestion d'un cluster par Astra Control :

Développez pour les étapes

i. Créer un `ClusterRole` fichier appelé, par exemple, `astra-admin-account.yaml`.

Ajustez le nom et l'espace de noms selon vos besoins. Si des modifications sont apportées ici, vous devez appliquer les mêmes modifications dans les étapes suivantes.

```
<strong>astra-admin-account.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:

# Get, List, Create, and Update all resources
# Necessary to backup and restore all resources in an app
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - get
  - list
  - create
  - patch

# Delete Resources
# Necessary for in-place restore and AppMirror failover
- apiGroups:
  - ""
  - apps
  - autoscaling
  - batch
  - crd.projectcalico.org
  - extensions
  - networking.k8s.io
  - policy
  - rbac.authorization.k8s.io
  - snapshot.storage.k8s.io
  - trident.netapp.io
  resources:
  - configmaps
  - cronjobs
  - daemonsets
```

```
    - deployments
    - horizontalpodautoscalers
    - ingresses
    - jobs
    - namespaces
    - networkpolicies
    - persistentvolumeclaims
    - poddisruptionbudgets
    - pods
    - podtemplates
    - podsecuritypolicies
    - replicasets
    - replicationcontrollers
    - replicationcontrollers/scale
    - rolebindings
    - roles
    - secrets
    - serviceaccounts
    - services
    - statefulsets
    - tridentmirrorrelationships
    - tridentsnapshotinfos
    - volumesnapshots
    - volumesnapshotcontents
    verbs:
    - delete

  # Watch resources
  # Necessary to monitor progress
  - apiGroups:
    - ""
    resources:
    - pods
    - replicationcontrollers
    - replicationcontrollers/scale
    verbs:
    - watch

  # Update resources
  - apiGroups:
    - ""
    - build.openshift.io
    - image.openshift.io
    resources:
    - builds/details
    - replicationcontrollers
```

```
  - replicationcontrollers/scale
  - imagestreams/layers
  - imagestreamtags
  - imagetags
  verbs:
  - update

# Use PodSecurityPolicies
- apiGroups:
  - extensions
  - policy
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

ii. (Pour les clusters OpenShift uniquement) Ajouter les éléments suivants à la fin du `astra-admin-account.yaml` ou après `# Use PodSecurityPolicies` section :

```
# OpenShift security
- apiGroups:
  - security.openshift.io
  resources:
  - securitycontextconstraints
  verbs:
  - use
```

iii. Appliquer le rôle de cluster :

```
kubectl apply -f astra-admin-account.yaml
```

- **Rôle de cluster étendu** : ce rôle contient des autorisations étendues pour un cluster devant être géré par Astra Control. Vous pouvez utiliser ce rôle si vous utilisez plusieurs contextes et que vous ne pouvez pas utiliser le kubeconfig Astra Control par défaut configuré lors de l'installation, ou si un rôle limité avec un seul contexte ne fonctionnera pas dans votre environnement :

> Les éléments suivants `ClusterRole` Les étapes constituent un exemple Kubernetes général. Pour des instructions spécifiques à votre environnement, reportez-vous à la documentation de votre distribution Kubernetes.

Développez pour les étapes

i. Créer un `ClusterRole` fichier appelé, par exemple, `astra-admin-account.yaml`.

   Ajustez le nom et l'espace de noms selon vos besoins. Si des modifications sont apportées ici, vous devez appliquer les mêmes modifications dans les étapes suivantes.

```
<strong>astra-admin-account.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'
```

ii. Appliquer le rôle de cluster :

```
kubectl apply -f astra-admin-account.yaml
```

3. Créer la liaison de rôle cluster pour le rôle cluster vers le compte de service :

   a. Créer un `ClusterRoleBinding` fichier appelé `astracontrol-clusterrolebinding.yaml`.

      Ajustez les noms et espaces de noms modifiés lors de la création du compte de service, le cas échéant.

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

+

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astracontrol-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: astra-admin-account
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

a. Appliquer la liaison de rôle de cluster :

```
kubectl apply -f astracontrol-clusterrolebinding.yaml
```

4. Créez et appliquez le secret de jeton :

a. Créez un fichier secret de jeton appelé `secret-astracontrol-service-account.yaml`.

```
<strong>secret-astracontrol-service-account.yaml</strong>
```

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: secret-astracontrol-service-account
  namespace: default
  annotations:
    kubernetes.io/service-account.name: "astracontrol-service-
account"
type: kubernetes.io/service-account-token
```

b. Appliquer le secret de jeton :

```
kubectl apply -f secret-astracontrol-service-account.yaml
```

5. Ajoutez le secret de jeton au compte de service en ajoutant son nom au `secrets` tableau (dernière ligne de l'exemple suivant) :

```
kubectl edit sa astracontrol-service-account
```

```
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-48xhx
kind: ServiceAccount
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"v1","kind":"ServiceAccount","metadata":{"annotations":{},
"name":"astracontrol-service-account","namespace":"default"}}
  creationTimestamp: "2023-06-14T15:25:45Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "2767069"
  uid: 2ce068c4-810e-4a96-ada3-49cbf9ec3f89
secrets:
- name: astracontrol-service-account-dockercfg-48xhx
<strong>- name: secret-astracontrol-service-account</strong>
```

6. Indiquez les secrets du compte de service, en les remplaçant `<context>` avec le contexte approprié pour votre installation :

```
kubectl get serviceaccount astracontrol-service-account --context
<context> --namespace default -o json
```

La fin de la sortie doit ressembler à ce qui suit :

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-48xhx"},
{ "name": "secret-astracontrol-service-account"}
]
```

Les indices pour chaque élément dans `secrets` la matrice commence par 0. Dans l'exemple ci-dessus, l'index de `astracontrol-service-account-dockercfg-48xhx` serait 0 et l'index pour `secret-astracontrol-service-account` serait 1. Dans votre sortie, notez le numéro d'index du compte de service secret. Vous aurez besoin de ce numéro d'index à l'étape suivante.

7. Générez le kubeconfig comme suit :

   a. Créer un `create-kubeconfig.sh` fichier. Remplacement `TOKEN_INDEX` au début du script suivant avec la valeur correcte.

```
<strong>create-kubeconfig.sh</strong>
```

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
```

```
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token
-user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

b. Source des commandes à appliquer à votre cluster Kubernetes.

```
source create-kubeconfig.sh
```

8. (Facultatif) Renommer le kubeconfig pour nommer votre cluster.

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

Et la suite ?

Maintenant que vous avez vérifié que les conditions préalables sont remplies, vous êtes prêt à ajouter un cluster.

Ajouter un cluster

Pour commencer à gérer vos applications, ajoutez un cluster Kubernetes et gérez-le comme une ressource de calcul. Il faut ajouter un cluster pour découvrir vos applications Kubernetes pour Astra Control Center.

> Nous vous recommandons de gérer le cluster qu'Astra Control Center déploie en premier avant d'ajouter d'autres clusters à Astra Control Center. La gestion du cluster initial est nécessaire pour envoyer les données Kubemetrics et les données associées au cluster pour les mesures et le dépannage.

Avant de commencer

- Avant d'ajouter un cluster, vérifiez et effectuez les opérations nécessaires tâches préalables.

Étapes

1. Naviguer à partir du menu Tableau de bord ou clusters :
   - Dans **Dashboard**, sélectionnez **Add** dans le volet clusters.
   - Dans la zone de navigation de gauche, sélectionnez **clusters**, puis **Ajouter un cluster** à partir de la

page clusters.

2. Dans la fenêtre **Ajouter un cluster** qui s'ouvre, chargez un `kubeconfig.yaml` classez le contenu d'un `kubeconfig.yaml` fichier.

> Le `kubeconfig.yaml` le fichier doit inclure **uniquement les informations d'identification du cluster pour un cluster**.

> Si vous créez la vôtre `kubeconfig` fichier, vous ne devez définir que **un** élément de contexte dans celui-ci. Reportez-vous à la section "Documentation Kubernetes" pour plus d'informations sur la création `kubeconfig` fichiers. Si vous avez créé un kubeconfig pour un rôle de cluster limité à l'aide de le processus ci-dessus, assurez-vous de télécharger ou de coller ce kubeconfig dans cette étape.

3. Indiquez un nom d'identification. Par défaut, le nom des identifiants est automatiquement renseigné comme nom du cluster.
4. Sélectionnez **Suivant**.
5. Sélectionnez la classe de stockage par défaut à utiliser pour ce cluster Kubernetes et sélectionnez **Suivant**.

> Vous devez sélectionner une classe de stockage Astra Trident reposant sur le stockage ONTAP.

6. Passez en revue les informations, et si tout semble bien, sélectionnez **Ajouter**.

Résultat

Le cluster passe à l'état **découverte**, puis passe à **sain**. Vous gérez maintenant le cluster avec Astra Control Center.

> Une fois que vous avez ajouté un cluster à gérer dans Astra Control Center, le déploiement de l'opérateur de surveillance peut prendre quelques minutes. En attendant, l'icône notification devient rouge et consigne un événement **échec de la vérification de l'état de l'agent de surveillance**. Vous pouvez ignorer cela car le problème résout lorsque le centre de contrôle Astra obtient le statut correct. Si le problème ne résout pas le problème en quelques minutes, accédez au cluster, puis exécutez-le `oc get pods -n netapp-monitoring` comme point de départ. Vous devrez consulter les journaux de l'opérateur de surveillance pour déboguer le problème.

Activez l'authentification sur le back-end de stockage ONTAP

ASTRA Control Center offre deux modes d'authentification d'un backend ONTAP :

- **Authentification basée sur les informations d'identification** : le nom d'utilisateur et le mot de passe d'un utilisateur ONTAP avec les autorisations requises. Vous devez utiliser un rôle de connexion de sécurité prédéfini, tel que admin ou vsadmin, pour assurer une compatibilité maximale avec les versions de ONTAP.
- **Authentification basée sur un certificat** : Astra Control Center peut également communiquer avec un cluster ONTAP à l'aide d'un certificat installé sur le back-end. Vous devez utiliser le certificat client, la clé et le certificat de l'autorité de certification approuvée, le cas échéant (recommandé).

Vous pouvez par la suite mettre à jour les systèmes back-end existants pour passer d'un type

d'authentification à une autre. Une seule méthode d'authentification est prise en charge à la fois.

Activer l'authentification basée sur les informations d'identification

ASTRA Control Center requiert les identifiants d'un cluster-scoped `admin` Pour communiquer avec le backend ONTAP. Vous devez utiliser des rôles standard prédéfinis, tels que `admin`. La compatibilité avec les futures versions d'ONTAP qui pourraient exposer les API de fonctionnalités à utiliser dans les futures versions d'Astra Control Center est ainsi garantie.

> Un rôle de connexion de sécurité personnalisé peut être créé et utilisé avec Astra Control Center, mais il n'est pas recommandé.

Un exemple de définition de back-end se présente comme suit :

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "admin",
  "password": "secret"
}
```

La définition du back-end est le seul endroit où les informations d'identification sont stockées en texte brut. La création ou la mise à jour d'un back-end est la seule étape qui nécessite la connaissance des informations d'identification. Il s'agit donc d'une opération réservée à l'administrateur du stockage ou de Kubernetes.

Activer l'authentification basée sur certificat

ASTRA Control Center peut utiliser des certificats pour communiquer avec les systèmes back-end ONTAP, nouveaux et existants. Vous devez entrer les informations suivantes dans la définition du back-end.

- `clientCertificate`: Certificat client.
- `clientPrivateKey`: Clé privée associée.
- `trustedCACertificate`: Certificat de l'autorité de certification approuvée. Si vous utilisez une autorité de certification approuvée, ce paramètre doit être fourni. Ceci peut être ignoré si aucune autorité de certification approuvée n'est utilisée.

Vous pouvez utiliser l'un des types de certificats suivants :

- Certificat auto-signé
- Certificat tiers

Activez l'authentification avec un certificat auto-signé

Un flux de travail type comprend les étapes suivantes.

Étapes

1. Générez un certificat client et une clé. Lors de la génération, définissez le nom commun (CN) sur l'utilisateur ONTAP pour s'authentifier en tant que.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=<common-name>"
```

2. Installez le certificat client de type `client-ca` Et sur le cluster ONTAP.

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

3. Vérifiez que le rôle de connexion de sécurité ONTAP prend en charge la méthode d'authentification par certificat.

```
security login create -user-or-group-name vsadmin -application ontapi
-authentication-method cert -vserver <vserver-name>
security login create -user-or-group-name vsadmin -application http
-authentication-method cert -vserver <vserver-name>
```

4. Tester l'authentification à l'aide du certificat généré. Remplacer <LIF> et <vserver name> de ONTAP par l'IP et le nom du SVM de la LIF de gestion. Vous devez vous assurer que le LIF a sa politique de service définie sur `default-data-management`.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns=http://www.netapp.com/filer/admin version="1.21" vfiler="<vserver-
name>"><vserver-get></vserver-get></netapp>
```

5. À l'aide des valeurs obtenues à l'étape précédente, ajoutez le back-end de stockage dans l'interface utilisateur d'Astra Control Center.

Activez l'authentification à l'aide d'un certificat tiers

Si vous disposez d'un certificat tiers, vous pouvez configurer l'authentification basée sur un certificat à l'aide de ces étapes.

Étapes

1. Générer la clé privée et la RSC :

```
openssl req -new -newkey rsa:4096 -nodes -sha256 -subj "/" -outform pem
-out ontap_cert_request.csr -keyout ontap_cert_request.key -addext
"subjectAltName = DNS:<ONTAP_CLUSTER_FQDN_NAME>,IP:<ONTAP_MGMT_IP>”
```

2. Transmettez la RSC à l’autorité de certification Windows (autorité de certification tierce) et émettez le certificat signé.
3. Téléchargez le certificat signé et nommez-le « ontap_signed_cert.crt ».
4. Exportez le certificat racine à partir de l’autorité de certification Windows (autorité de certification tierce).
5. Nommez ce fichier `ca_root.crt`

   Vous disposez maintenant des trois fichiers suivants :

   - **Clé privée** : `ontap_signed_request.key` (Il s’agit de la clé correspondante pour le certificat de serveur dans ONTAP. Elle est nécessaire lors de l’installation du certificat du serveur.)
   - **Certificat signé**: `ontap_signed_cert.crt` (Il s’agit également du *certificat de serveur* dans ONTAP.)
   - **Certificat CA racine** : `ca_root.crt` (Il s’agit également du certificat *Server-ca* dans ONTAP.)
6. Installez ces certificats dans ONTAP. Générer et installer `server` et `server-ca` Certificats sur ONTAP.

Développez pour Sample.yaml

```
# Copy the contents of ca_root.crt and use it here.

security certificate install -type server-ca

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----


You should keep a copy of the CA-signed digital certificate for
future reference.

The installed certificate's CA and serial number for reference:

CA:
serial:

The certificate's generated name for reference:


===

# Copy the contents of ontap_signed_cert.crt and use it here. For
key, use the contents of ontap_cert_request.key file.
security certificate install -type server
Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----

Please enter Private Key: Press <Enter> when done

-----BEGIN PRIVATE KEY-----
<private key details>
-----END PRIVATE KEY-----

Enter certificates of certification authorities (CA) which form the
certificate chain of the server certificate. This starts with the
issuing CA certificate of the server certificate and can range up to
the root CA certificate.
Do you want to continue entering root and/or intermediate
```

```
certificates {y|n}: n

The provided certificate does not have a common name in the subject
field.
Enter a valid common name to continue installation of the
certificate: <ONTAP_CLUSTER_FQDN_NAME>

You should keep a copy of the private key and the CA-signed digital
certificate for future reference.
The installed certificate's CA and serial number for reference:
CA:
serial:
The certificate's generated name for reference:


==
# Modify the vserver settings to enable SSL for the installed
certificate

ssl modify -vserver <vserver_name> -ca <CA>  -server-enabled true
-serial <serial number>       (security ssl modify)

==
# Verify if the certificate works fine:

openssl s_client -CAfile ca_root.crt -showcerts -servername server
-connect <ONTAP_CLUSTER_FQDN_NAME>:443
CONNECTED(00000005)
depth=1 DC = local, DC = umca, CN = <CA>
verify return:1
depth=0
verify return:1
write W BLOCK
---
Certificate chain
0 s:
   i:/DC=local/DC=umca/<CA>

-----BEGIN CERTIFICATE-----
<Certificate details>
```

7. Créez le certificat client pour le même hôte pour la communication sans mot de passe. ASTRA Control Center utilise ce processus pour communiquer avec ONTAP.
8. Générer et installer les certificats client sur ONTAP :

Développez pour Sample.yaml

```
# Use /CN=admin or use some other account which has privileges.
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout
ontap_test_client.key -out ontap_test_client.pem -subj "/CN=admin"

Copy the content of ontap_test_client.pem file and use it in the
below command:
security certificate install -type client-ca -vserver <vserver_name>

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<Certificate details>
-----END CERTIFICATE-----

You should keep a copy of the CA-signed digital certificate for
future reference.
The installed certificate’s CA and serial number for reference:

CA:
serial:
The certificate’s generated name for reference:


==

ssl modify -vserver <vserver_name> -client-enabled true
(security ssl modify)

# Setting permissions for certificates
security login create -user-or-group-name admin -application ontapi
-authentication-method cert -role admin -vserver <vserver_name>

security login create -user-or-group-name admin -application http
-authentication-method cert -role admin -vserver <vserver_name>

==

#Verify passwordless communication works fine with the use of only
certificates:

curl --cacert ontap_signed_cert.crt  --key ontap_test_client.key
--cert ontap_test_client.pem
https://<ONTAP_CLUSTER_FQDN_NAME>/api/storage/aggregates
{
```

```
"records": [
{
"uuid": "f84e0a9b-e72f-4431-88c4-4bf5378b41bd",
"name": "<aggr_name>",
"node": {
"uuid": "7835876c-3484-11ed-97bb-d039ea50375c",
"name": "<node_name>",
"_links": {
"self": {
"href": "/api/cluster/nodes/7835876c-3484-11ed-97bb-d039ea50375c"
}
}
},
"_links": {
"self": {
"href": "/api/storage/aggregates/f84e0a9b-e72f-4431-88c4-
4bf5378b41bd"
}
}
}
],
"num_records": 1,
"_links": {
"self": {
"href": "/api/storage/aggregates"
}
}
}%
```


9. Ajoutez le système back-end de stockage dans l'interface utilisateur d'Astra Control Center et fournissez les valeurs suivantes :
   - **Certificat client** : ontap_test_client.pem
   - **Clé privée** : ontap_test_client.key
   - **Certificat CA de confiance** : ontap_signed_cert.crt

Ajout d'un système back-end

Vous pouvez ajouter un système de stockage back-end ONTAP à Astra Control Center pour gérer ses ressources.

La gestion des clusters de stockage d'Astra Control en tant que backend de stockage vous permet d'obtenir des liens entre les volumes persistants (PVS) et le back-end de stockage, ainsi que des metrics de stockage supplémentaires.

Après avoir configuré les informations d'identification ou d'authentification de certificat, vous pouvez ajouter un système back-end de stockage ONTAP existant à Astra Control Center pour gérer ses ressources.

Étapes

1. Dans la zone de navigation gauche du tableau de bord, sélectionnez **Backends**.
2. Sélectionnez **Ajouter**.
3. Dans la section utiliser existant de la page Ajouter un back-end de stockage, sélectionnez **ONTAP**.
4. Sélectionnez l'une des options suivantes :
   - **Utiliser les informations d'identification de l'administrateur** : saisissez l'adresse IP de gestion du cluster ONTAP et les informations d'identification de l'administrateur. Les identifiants doivent être identifiants au niveau du cluster.

     > L'utilisateur dont vous saisissez ici les informations d'identification doit disposer du `ontapi` Méthode d'accès de connexion utilisateur activée dans ONTAP System Manager sur le cluster ONTAP. Si vous prévoyez d'utiliser la réplication SnapMirror, appliquez les identifiants de l'utilisateur au rôle « admin », qui dispose des méthodes d'accès `ontapi` et `http`, Sur les clusters ONTAP source et destination. Reportez-vous à la section "Gérer les comptes utilisateur dans la documentation ONTAP" pour en savoir plus.

   - **Utiliser un certificat**: Télécharger le certificat `.pem` fichier, la clé de certificat `.key` et éventuellement le fichier de l'autorité de certification.
5. Sélectionnez **Suivant**.
6. Confirmez les détails du back-end et sélectionnez **gérer**.

Résultat

Le back-end s'affiche dans le `online` état dans la liste avec des informations récapitulatives.

> Vous devrez peut-être actualiser la page pour que le back-end apparaisse.

Ajouter un godet

Vous pouvez ajouter un compartiment à l'aide de l'interface utilisateur Astra Control ou "API de contrôle Astra". Il est essentiel d'ajouter des fournisseurs de compartiments de stockage objet pour sauvegarder les applications et le stockage persistant ou pour cloner les applications entre les clusters. Astra Control stocke les sauvegardes ou les clones dans les compartiments de magasin d'objets que vous définissez.

Si vous clonez la configuration de vos applications et le stockage persistant vers le même cluster, il n'est pas nécessaire d'utiliser un compartiment dans Astra Control. La fonctionnalité de copie Snapshot des applications ne nécessite pas de compartiment.

Avant de commencer

- Compartiment accessible depuis vos clusters gérés par Astra Control Center.
- Identifiants pour le compartiment.
- Un godet des types suivants :
  - NetApp ONTAP S3
  - NetApp StorageGRID S3
  - Microsoft Azure
  - S3 générique

Amazon Web Services (AWS) et Google Cloud Platform (GCP) utilisent le type de compartiment S3 générique.

Bien qu'Astra Control Center prenne en charge Amazon S3 en tant que fournisseur de compartiments génériques, Astra Control Center peut ne pas prendre en charge tous les fournisseurs de magasins d'objets qui affirment la prise en charge d'Amazon S3.

Étapes

1. Dans la zone de navigation de gauche, sélectionnez **godets**.
2. Sélectionnez **Ajouter**.
3. Sélectionner le type de godet.

   Lorsque vous ajoutez un compartiment, sélectionnez le fournisseur approprié et fournissez les identifiants appropriés pour ce fournisseur. Par exemple, l'interface utilisateur accepte NetApp ONTAP S3 comme type et accepte les identifiants StorageGRID. Toutefois, toutes les futures sauvegardes et restaurations des applications à l'aide de ce compartiment échoueront.

4. Saisissez un nom de compartiment existant et une description facultative.

   Le nom et la description du compartiment apparaissent comme un emplacement de sauvegarde que vous pouvez choisir plus tard lors de la création d'une sauvegarde. Ce nom apparaît également lors de la configuration de la règle de protection.

5. Entrez le nom ou l'adresse IP du terminal S3.
6. Sous **Sélectionner les informations d'identification**, choisissez l'onglet **Ajouter** ou **utiliser l'onglet existant**.
   - Si vous avez choisi **Ajouter**:
     i. Saisissez un nom pour l'identifiant qui le distingue des autres identifiants dans Astra Control.
     ii. Saisissez l'ID d'accès et la clé secrète en collant le contenu dans le presse-papiers.
   - Si vous avez choisi **utiliser existant**:
     i. Sélectionnez les informations d'identification existantes à utiliser avec le compartiment.
7. Sélectionnez `Add`.

   Lorsque vous ajoutez un godet, Astra Control marque un godet avec l'indicateur de compartiment par défaut. Le premier compartiment que vous créez devient le compartiment par défaut. Au fur et à mesure que vous ajoutez des compartiments, vous pourrez décider plus tard "définir un autre compartiment par défaut".

Et la suite ?

Maintenant que vous êtes connecté et que vous avez ajouté des clusters à Astra Control Center, vous êtes prêt à utiliser les fonctionnalités de gestion des données applicatives d'Astra Control Center.

- "Gérez les utilisateurs et les rôles locaux"
- "Commencez à gérer les applications"

- "Protégez vos applications"
- "Gérer les notifications"
- "Connectez-vous à Cloud Insights"
- "Ajouter un certificat TLS personnalisé"
- "Modifiez la classe de stockage par défaut"

Trouvez plus d'informations

- "Utilisez l'API de contrôle Astra"
- "Problèmes connus"

Foire aux questions pour Astra Control Center

Cette FAQ peut vous aider si vous cherchez juste une réponse rapide à une question.

Présentation

Les sections suivantes fournissent des réponses à des questions supplémentaires que vous pourriez vous poser lorsque vous utilisez le centre de contrôle Astra. Pour plus de précisions, veuillez contacter astra.feedback@netapp.com

Accès au centre de contrôle Astra

Qu'est-ce que l'URL de contrôle Astra?

Astra Control Center utilise l'authentification locale et une URL spécifique à chaque environnement.

Pour l'URL, dans un navigateur, entrez le nom de domaine complet (FQDN) que vous avez défini dans le champ spec.astraAddress du fichier de ressource personnalisée astra_control_Center.yaml lorsque vous avez installé Astra Control Center. L'e-mail est la valeur que vous avez définie dans le champ spec.email de l'astra_Control_Center.yaml CR.

Licences

J'utilise une licence d'évaluation. Comment passer à la licence complète?

Vous pouvez facilement choisir une licence complète en obtenant le fichier de licence NetApp (NLF) auprès de NetApp.

Étapes

1. Dans le menu de navigation de gauche, sélectionnez **compte** > **Licence**.
2. Dans la vue d'ensemble de la licence, à droite des informations de licence, sélectionnez le menu Options.
3. Sélectionnez **remplacer**.
4. Naviguez jusqu'au fichier de licence que vous avez téléchargé et sélectionnez **Ajouter**.

J'utilise une licence d'évaluation. Puis-je toujours gérer les applications ?

Oui, vous pouvez tester la fonctionnalité de gestion des applications avec une licence d'évaluation (y compris

la licence d'évaluation intégrée installée par défaut). Il n'y a pas de différence entre les capacités ou les fonctionnalités d'une licence d'évaluation et d'une licence complète; la licence d'évaluation a simplement une durée de vie plus courte. Reportez-vous à la section "Licences" pour en savoir plus.

Enregistrement des clusters Kubernetes

J'ai besoin d'ajouter des nœuds workers à mon cluster Kubernetes après avoir ajouté Astra Control. Que dois-je faire?

De nouveaux nœuds workers peuvent être ajoutés aux pools existants. Elles seront automatiquement découvertes par Astra Control. Si les nouveaux nœuds ne sont pas visibles dans Astra Control, vérifiez si les nouveaux nœuds de travail exécutent le type d'image pris en charge. Vous pouvez également vérifier l'état de santé des nouveaux nœuds workers à l'aide de la `kubectl get nodes` commande.

Comment puis-je dégérer correctement un cluster?

1. "Gérez les applications avec Astra Control".
2. "Dégérer le cluster à partir d'Astra Control".

Que se passe-t-il pour mes applications et données après avoir retiré le cluster Kubernetes d'Astra Control?

La suppression d'un cluster d'Astra Control ne modifie pas la configuration du cluster (applications et stockage persistant). Toute restauration de snapshots ou de sauvegardes Astra Control effectuée sur ce cluster sera indisponible. Les sauvegardes de stockage persistant créées par Astra Control restent dans le contrôle d'Astra, mais elles sont indisponibles pour les restaurations.

> Retirez toujours un cluster d'Astra Control avant de le supprimer par d'autres méthodes. La suppression d'un cluster à l'aide d'un autre outil alors qu'il est toujours géré par Astra Control peut causer des problèmes pour votre compte Astra Control.

NetApp Astra Trident est-il automatiquement désinstallé d'un cluster lorsque je ne le gère pas ?
Lorsque vous dégérez un cluster depuis Astra Control Center, Astra Trident n'est pas automatiquement désinstallé du cluster. Pour désinstaller Astra Trident, vous devez "Suivez ces étapes dans la documentation d'Astra Trident".

La gestion des applications

Astra Control peut-il déployer une application?

Astra Control ne déploie pas d'applications. Les applications doivent être déployées en dehors d'Astra Control.

Que se passe-t-il pour les applications après que je les ai cessent de les gérer à partir d'Astra Control?

Toutes les sauvegardes ou tous les instantanés existants seront supprimés. Les applications et les données restent disponibles. Les opérations de gestion des données ne seront pas disponibles pour les applications non gérées ni pour les sauvegardes ou snapshots qui y appartiennent.

Astra Control peut-il gérer une application qui se trouve sur un système de stockage autre que NetApp?

Non Astra Control peut découvrir des applications qui utilisent un stockage autre que NetApp, mais il ne peut pas gérer une application qui utilise un stockage non NetApp.

Devrais-je gérer Astra Control lui-même?

ASTRA Control Center n’est pas affiché par défaut en tant qu’application que vous pouvez gérer, mais vous pouvez sauvegarder et restaurer une instance Astra Control Center à l’aide d’une autre instance Astra Control Center.

Les pods défectueux affectent-ils la gestion des applications ?

Non, l’état des pods n’a pas d’impact sur la gestion des applications.

Les opérations de gestion des données

Mon application utilise plusieurs PVS. ASTRA Control prendra-t-il des snapshots et des sauvegardes de ces volumes persistants ?

Oui. Une opération d’instantané sur une application par Astra Control inclut un instantané de tous les volumes persistants liés aux demandes de volume persistant de l’application.

Puis-je gérer les instantanés pris par Astra Control directement via une interface ou un stockage objet différent?

Non Les copies Snapshot et les sauvegardes effectuées par Astra Control ne peuvent être gérées qu’avec Astra Control.