Déployez-le avec l'opérateur Trident

Astra Trident

NetApp
April 16, 2024

This PDF was generated from https://docs.netapp.com/fr-fr/trident-2210/trident-get-started/kubernetes-customize-deploy.html on April 16, 2024. Always check docs.netapp.com for the latest.

Sommaire

Déployez-le avec l'opérateur Trident . . . 1
Découvrez Astra Trident 22.10, un document essentiel . . . 1
Options de déploiement de l'opérateur Trident . . . 1
Vérifiez les prérequis . . . 1
Déployer l'opérateur Trident et installer Astra Trident à l'aide de Helm . . . 2
Déployez l'opérateur Trident manuellement et installez Trident . . . 3
Personnalisez le déploiement des opérateurs Trident . . . 8

Déployez-le avec l'opérateur Trident

Vous pouvez déployer Astra Trident avec l'opérateur Trident.

Découvrez Astra Trident 22.10, un document essentiel

Vous devez lire les informations essentielles suivantes avant de passer à Astra Trident 22.10.

, la protection des données essentielles d'Astra Trident 22.10

- Kubernetes 1.25 est désormais pris en charge par Trident. Vous devez effectuer la mise à niveau vers Astra Trident 22.10 avant la mise à niveau vers Kubernetes 1.25.
- Astra Trident applique désormais rigoureusement la configuration des chemins d'accès multiples dans les environnements SAN, avec la valeur recommandée de `find_multipaths: no` dans le fichier multipath.conf.

  Utilisation d'une configuration sans chemins d'accès multiples ou de l'utilisation de `find_multipaths: yes` ou `find_multipaths: smart` la valeur du fichier multipath.conf entraînera des échecs de montage. Trident a recommandé l'utilisation de `find_multipaths: no` depuis la version 21.07.

Options de déploiement de l'opérateur Trident

Vous pouvez déployer l'opérateur Trident de deux manières :

- Avec Trident "Graphique Helm": Le graphique Helm déploie l'opérateur Trident et installe Trident en une étape.
- Manuellement : Trident fournit un fichier qui peut être utilisé pour installer l'opérateur et créer des objets associés.
  - Pour les clusters exécutant Kubernetes 1.24 ou version inférieure, utilisez "bundle_pre_1_25.yaml".
  - Pour les clusters qui exécutent Kubernetes 1.25 ou version supérieure, utilisez "bundle_post_1_25.yaml".

Si vous ne vous êtes pas déjà familiarisé avec le "concepts de base", c'est le moment idéal pour le faire.

Vérifiez les prérequis

Pour déployer Astra Trident, les prérequis suivants doivent être respectés :

- Vous disposez de privilèges complets pour un cluster Kubernetes pris en charge exécutant une version de Kubernetes prise en charge. Vérifiez le "de formation".
- Vous avez accès à un système de stockage NetApp pris en charge.
- Vous avez la possibilité de monter des volumes à partir de tous les nœuds workers Kubernetes.
- Vous avez un hôte Linux avec `kubectl` (ou `oc`, Si vous utilisez OpenShift) installé et configuré pour gérer le cluster Kubernetes que vous souhaitez utiliser.
- Vous avez défini le `KUBECONFIG` Variable d'environnement qui pointe vers votre configuration de cluster

Kubernetes.

- Vous avez activé "Portails requis par Astra Trident".
- Si vous utilisez Kubernetes avec Docker Enterprise, "Suivez les étapes indiquées pour activer l'accès à l'interface de ligne de commande".

Vous avez tout ça ? Parfait ! Nous allons commencer.

Déployer l'opérateur Trident et installer Astra Trident à l'aide de Helm

Effectuer les étapes énumérées pour déployer l'opérateur Trident à l'aide de Helm.

Ce dont vous avez besoin

En plus des prérequis répertoriés ci-dessus, pour déployer l'opérateur Trident à l'aide de Helm, vous devez :

- A "Version Kubernetes prise en charge"
- Version 3 de Helm

Étapes

1. Ajouter le référentiel Helm de Trident :

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-chart
```

2. Utilisez le `helm install` et spécifiez un nom pour votre déploiement. Voir l'exemple suivant :

```
helm install <name> netapp-trident/trident-operator --version 22.10.0
--create-namespace --namespace <trident-namespace>
```

> Si vous avez déjà créé un namespace pour Trident, le `--create-namespace` le paramètre ne crée pas d'espace de noms supplémentaire.

Il existe deux façons de passer les données de configuration au cours de l'installation :

- `--values` (ou `-f`) : Spécifiez un fichier YAML avec les remplacements. Ceci peut être spécifié plusieurs fois et le fichier le plus à droite sera prioritaire.
- `--set`: Spécifiez les remplacements sur la ligne de commande.

Par exemple, pour modifier la valeur par défaut de `debug`, exécutez ce qui suit `--set` commande :

```
helm install <name> netapp-trident/trident-operator --version 22.10.0
--create-namespace --namespace --set tridentDebug=true
```

Le `values.yaml` Le fichier, qui fait partie du graphique Helm, fournit la liste des clés et leurs valeurs par défaut.

`helm list` affiche des informations détaillées sur l'installation, telles que nom, espace de noms, graphique, état, version de l'application, numéro de révision, etc.

Déployez l'opérateur Trident manuellement et installez Trident

Effectuer les étapes énumérées pour déployer manuellement l'opérateur Trident.

Étape 1 : qualifier le cluster Kubernetes

La première chose à faire est de se connecter à l'hôte Linux et de vérifier qu'il gère un *working*, "Cluster Kubernetes pris en charge" que vous disposez des privilèges nécessaires à.

> Avec OpenShift, utilisez `oc` au lieu de `kubectl` dans tous les exemples qui suivent, et connectez-vous en tant que **system:admin** en premier lieu en cours d'exécution `oc login -u system:admin` ou `oc login -u kube-admin`.

Pour vérifier votre version de Kubernetes, exécutez la commande suivante :

```
kubectl version
```

Pour vérifier si vous disposez des privilèges d'administrateur de cluster Kubernetes, exécutez la commande suivante :

```
kubectl auth can-i '*' '*' --all-namespaces
```

Pour vérifier si vous pouvez lancer un pod qui utilise une image de Docker Hub et atteindre votre système de stockage sur le réseau pod, exécutez la commande suivante :

```
kubectl run -i --tty ping --image=busybox --restart=Never --rm -- \
  ping <management IP>
```

Étape 2 : télécharger et configurer l'opérateur

> Depuis la version 21.01, l'opérateur de Trident se trouve dans le périmètre du cluster. Pour installer Trident, vous devez créer le `TridentOrchestrator` Définition de ressource personnalisée (CRD) et définition d'autres ressources. Vous devez effectuer ces étapes pour configurer l'opérateur avant de pouvoir installer Astra Trident.

1. Téléchargez et extrayez la dernière version du programme d'installation de Trident sur "La section *Assets* sur GitHub".

```
wget
https://github.com/NetApp/trident/releases/download/v22.10.0/trident-
installer-22.10.0.tar.gz
tar -xf trident-installer-22.10.0.tar.gz
cd trident-installer
```

2. Utilisez le manifeste CRD approprié pour créer le `TridentOrchestrator` CRD. Vous créez ensuite un `TridentOrchestrator` Ressource personnalisée plus tard pour instancier une installation par l'opérateur.

   Exécutez la commande suivante :

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. Après le `TridentOrchestrator` Le service CRD est créé et crée les ressources suivantes nécessaires au déploiement de l'opérateur :

   - Un ServiceAccount pour l'opérateur
   - A ClusterRole et ClusterRoleBinding au ServiceAccount
   - Une stratégie de sécurité de PodSecurityPolicy dédiée
   - L'opérateur lui-même

     Le programme d'installation Trident contient des manifestes pour définir ces ressources. Par défaut, l'opérateur est déployé dans le `trident` espace de noms. Si le `trident` l'espace de noms n'existe pas, utilisez le manifeste suivant pour en créer un.

```
kubectl apply -f deploy/namespace.yaml
```

4. Pour déployer l'opérateur dans un espace de noms autre que celui par défaut `trident` espace de noms, vous devez mettre à jour le `serviceaccount.yaml`, `clusterrolebinding.yaml` et `operator.yaml` manifeste et génère votre `bundle.yaml`.

   Exécutez la commande suivante pour mettre à jour les manifestes YAML et générer votre `bundle.yaml` à l'aide du `kustomization.yaml`:

```
kubectl kustomize deploy/ > deploy/bundle.yaml
```

   Exécutez la commande suivante pour créer les ressources et déployer l'opérateur :

```
kubectl create -f deploy/bundle.yaml
```

5. Pour vérifier l'état de l'opérateur après le déploiement, procédez comme suit :

```
kubectl get deployment -n <operator-namespace>

NAME                READY   UP-TO-DATE   AVAILABLE   AGE
trident-operator    1/1     1            1           3m
```

```
kubectl get pods -n <operator-namespace>

NAME                                READY   STATUS             RESTARTS
AGE
trident-operator-54cb664d-lnjxh     1/1     Running            0
3m
```

Le déploiement de l'opérateur a réussi à créer un pod exécuté sur l'un des nœuds worker de votre cluster.

> Il ne doit y avoir que **une instance** de l'opérateur dans un cluster Kubernetes. Ne créez pas plusieurs déploiements de l'opérateur Trident.

Étape 3 : créer `TridentOrchestrator` Et installer Trident

Vous êtes maintenant prêt à installer Astra Trident avec l'opérateur ! Cela nécessitera la création `TridentOrchestrator`. Le programme d'installation Trident est fourni avec des exemples de définitions à créer `TridentOrchestrator`. Cela déclenche une installation dans le `trident` espace de noms.

```
kubectl create -f deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created

kubectl describe torc trident
Name:        trident
Namespace:
Labels:      <none>
Annotations: <none>
API Version: trident.netapp.io/v1
Kind:        TridentOrchestrator
...
Spec:
  Debug:     true
  Namespace: trident
Status:
  Current Installation Params:
    IPv6:                      false
    Autosupport Hostname:
    Autosupport Image:         netapp/trident-autosupport:22.10
    Autosupport Proxy:
    Autosupport Serial Number:
    Debug:                     true
    Image Pull Secrets:
    Image Registry:
    k8sTimeout:           30
    Kubelet Dir:          /var/lib/kubelet
    Log Format:           text
    Silence Autosupport:  false
    Trident Image:        netapp/trident:21.04.0
  Message:                  Trident installed  Namespace:
trident
  Status:                   Installed
  Version:                  v21.04.0
Events:
    Type Reason Age From Message ---- ------ ---- ---- -------Normal
    Installing 74s trident-operator.netapp.io Installing Trident Normal
    Installed 67s trident-operator.netapp.io Trident installed
```

L'opérateur Trident vous permet de personnaliser l'installation d'Astra Trident à l'aide des attributs du `TridentOrchestrator` spécifications Voir "Personnalisez votre déploiement Trident".

Le statut de `TridentOrchestrator` Indique si l'installation a réussi et affiche la version de Trident installée.

État	Description
Installation	L'opérateur installe Astra Trident à l'aide de ce module `TridentOrchestrator` CR.
Installé	Astra Trident a été installé avec succès.
Désinstallation	L'opérateur désinstallant Astra Trident, car `spec.uninstall=true`.
Désinstallé	Astra Trident est désinstallé.
Échec	L'opérateur n'a pas pu installer, corriger, mettre à jour ou désinstaller Astra Trident. L'opérateur essaiera automatiquement de récupérer cet état. Si cet état persiste, vous devrez effectuer un dépannage.
Mise à jour	L'opérateur met à jour une installation existante.
Erreur	Le `TridentOrchestrator` n'est pas utilisé. Un autre existe déjà.

Pendant l'installation, l'état de `TridentOrchestrator` modifications de `Installing` à `Installed`. Si vous observez l' `Failed` statut et l'opérateur ne peut pas récupérer lui-même, il est recommandé de vérifier les journaux de l'opérateur. Voir la "dépannage" section.

Vous pouvez vérifier que l'installation d'Astra Trident est terminée en consultant les pods qui ont été créés :

```
kubectl get pod -n trident

NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-7d466bf5c7-v4cpw        5/5     Running   0           1m
trident-csi-mr6zc                   2/2     Running   0           1m
trident-csi-xrp7w                   2/2     Running   0           1m
trident-csi-zh2jt                   2/2     Running   0           1m
trident-operator-766f7b8658-ldzsv   1/1     Running   0           3m
```

Vous pouvez également utiliser `tridentctl` Pour vérifier la version d'Astra Trident installée.

```
./tridentctl -n trident version

+----------------+----------------+
| SERVER VERSION | CLIENT VERSION |
+----------------+----------------+
| 21.04.0        | 21.04.0        |
+----------------+----------------+
```

Maintenant, vous pouvez avancer et créer un back-end. Voir "tâches post-déploiement".

Pour résoudre les problèmes pendant le déploiement, reportez-vous au "dépannage" section.

Personnalisez le déploiement des opérateurs Trident

L'opérateur Trident vous permet de personnaliser l'installation d'Astra Trident à l'aide des attributs du `TridentOrchestrator` spécifications

Si vous voulez personnaliser l'installation au-delà de ce qui est `TridentOrchestrator` les arguments permettent, vous devez envisager d'utiliser `tridentctl` Pour générer des manifestes YAML personnalisés que vous pouvez modifier si nécessaire.

> `spec.namespace` est spécifié dans `TridentOrchestrator` Pour indiquer quel espace de noms d'Astra Trident est installé. Ce paramètre **ne peut pas être mis à jour après l'installation d'Astra Trident**. Pour tenter de le faire, le `TridentOrchestrator` statut pour passer à `Failed`. Astra Trident n'est pas conçu pour être migré entre les espaces de noms.

Options de configuration

Ce tableau est plus détaillé `TridentOrchestrator` attributs :

Paramètre	Description	Valeur par défaut
`namespace`	Espace de noms pour installer Astra Trident dans	« par défaut »
`debug`	Activez le débogage pour Astra Trident	faux
`windows`	Réglage sur `true` Active l'installation sur les nœuds de travail Windows.	faux
`IPv6`	Installez Astra Trident sur IPv6	faux
`k8sTimeout`	Délai d'expiration pour les opérations Kubernetes	30 secondes
`silenceAutosupport`	N'envoyez pas automatiquement des packs AutoSupport à NetApp	faux
`enableNodePrep`	Gérer automatiquement les dépendances des nœuds de travail (**BÊTA**)	faux
`autosupportImage`	Image conteneur pour la télémétrie AutoSupport	« netapp/trident-autosupport :22.10.0 »
`autosupportProxy`	Adresse/port d'un proxy pour l'envoi de télémétrie AutoSupport	"http://proxy.example.com:8888"
`uninstall`	Indicateur utilisé pour désinstaller Astra Trident	faux
`logFormat`	Format de connexion Astra Trident à utiliser [text,json]	« texte »

Paramètre	Description	Valeur par défaut
`tridentImage`	Image Astra Trident à installer	netapp/trident:21.04
`imageRegistry`	Chemin d'accès au registre interne, du format `<registry FQDN>[:port][/subpath]`	"k8s.gcr.io/sig-storage (k8s 1.19+) ou quay.io/k8scsi"
`kubeletDir`	Chemin d'accès au répertoire kubelet de l'hôte	"/var/lib/kubelet"
`wipeout`	Liste des ressources à supprimer pour effectuer la suppression complète d'Astra Trident	
`imagePullSecrets`	Secrets pour extraire des images d'un registre interne	
`controllerPluginNodeSelector`	Sélecteurs de nœud supplémentaires pour les pods exécutant le plug-in Trident Controller CSI. Suit le même format que pod.spec.nodeSelector.	Pas de valeur par défaut ; facultatif
`controllerPluginTolerations`	Remplace les tolérances pour les pods exécutant le plug-in Trident Controller CSI. Suit le même format que pod.spec.tolérances.	Pas de valeur par défaut ; facultatif
`nodePluginNodeSelector`	Sélecteurs de nœud supplémentaires pour les pods exécutant le plug-in Trident Node CSI. Suit le même format que pod.spec.nodeSelector.	Pas de valeur par défaut ; facultatif
`nodePluginTolerations`	Remplace les tolérances pour les pods exécutant le plug-in Trident Node CSI. Suit le même format que pod.spec.tolérances.	Pas de valeur par défaut ; facultatif

Pour plus d'informations sur le formatage des paramètres du pod, reportez-vous à la section "Attribution de pods aux nœuds".

Exemples de configurations

Vous pouvez utiliser les attributs mentionnés ci-dessus lors de la définition `TridentOrchestrator` pour personnaliser votre installation.

Exemple 1 : configuration personnalisée de base

Voici un exemple de configuration personnalisée de base.

```
cat deploy/crds/tridentorchestrator_cr_imagepullsecrets.yaml
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullSecrets:
  - thisisasecret
```

Exemple 2 : déploiement avec des sélecteurs de nœuds

Cet exemple illustre le déploiement de Trident avec des sélecteurs de nœud :

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  controllerPluginNodeSelector:
    nodetype: master
  nodePluginNodeSelector:
    storage: netapp
```

Exemple 3 : déploiement sur des nœuds de travail Windows

Cet exemple illustre le déploiement sur un nœud de travail Windows.

```
$ cat deploy/crds/tridentorchestrator_cr.yaml
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  windows: true
```

Informations sur le copyright

Copyright © 2024 NetApp, Inc. Tous droits réservés. Imprimé aux États-Unis. Aucune partie de ce document protégé par copyright ne peut être reproduite sous quelque forme que ce soit ou selon quelque méthode que ce soit (graphique, électronique ou mécanique, notamment par photocopie, enregistrement ou stockage dans un système de récupération électronique) sans l'autorisation écrite préalable du détenteur du droit de copyright.

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

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

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

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

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

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

Informations sur les marques commerciales

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