



Présentation des intégrations de stockage NetApp

NetApp Solutions

NetApp
April 26, 2024

Sommaire

- Présentation de l'intégration du stockage NetApp 1
 - Présentation de NetApp Astra Control Center 2
 - Présentation d'Astra Trident 30

Présentation de l'intégration du stockage NetApp

NetApp propose plusieurs produits pour orchestrer et gérer les données persistantes dans des environnements basés sur des conteneurs, tels que Red Hat OpenShift.



NetApp Astra Control propose un ensemble complet de services de gestion du stockage et des données respectueuse des applications pour les workloads Kubernetes avec état, optimisés par la technologie de protection des données NetApp. Astra Control Service est disponible pour la prise en charge des workloads avec état dans les déploiements Kubernetes cloud natifs. Le centre de contrôle Astra est disponible pour les workloads avec état dans les déploiements sur site, tels que Red Hat OpenShift. Pour en savoir plus, rendez-vous sur le site Web NetApp Astra Control ["ici"](#).

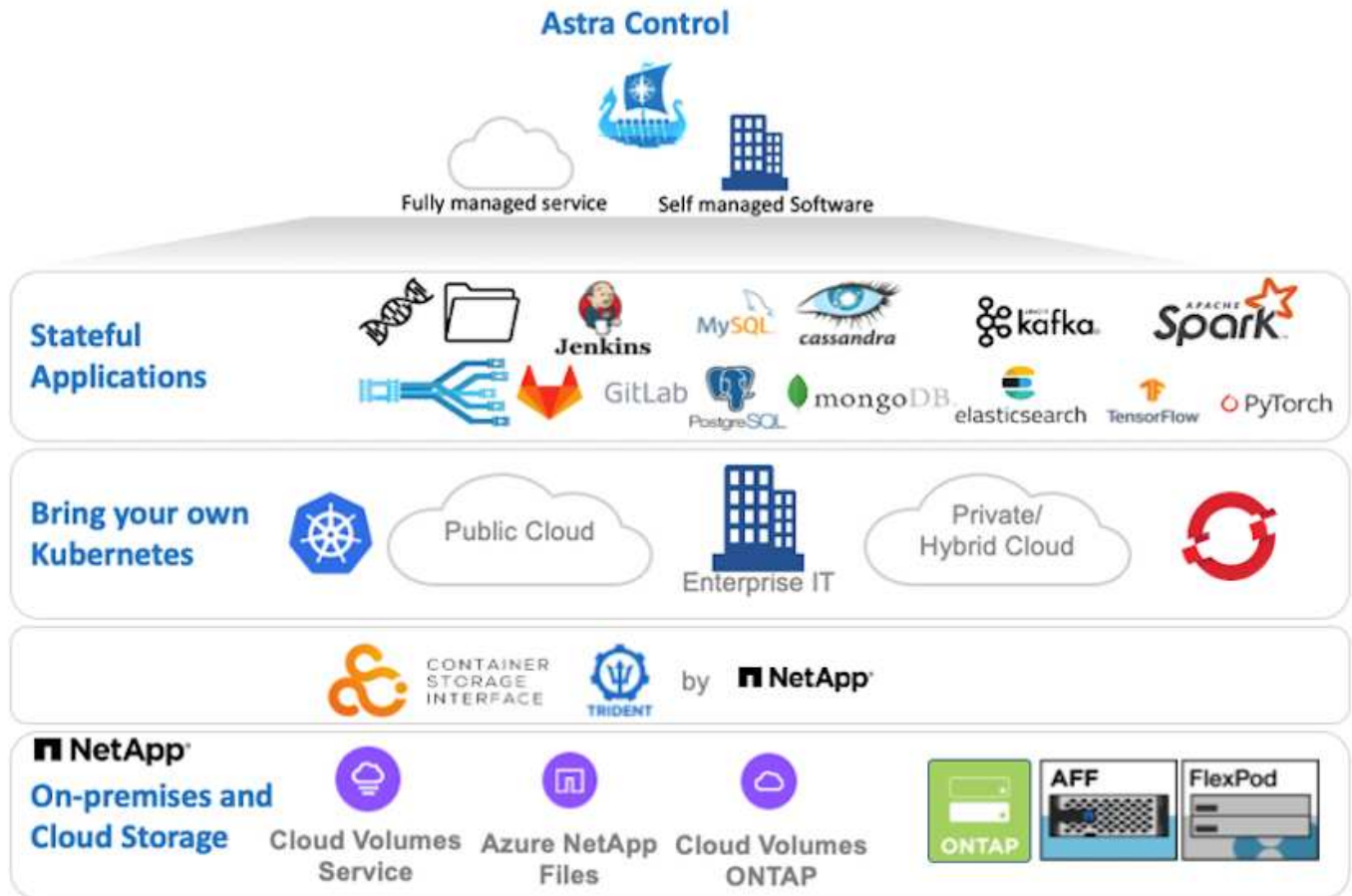
NetApp Astra Trident est un orchestrateur de stockage open source entièrement pris en charge pour les conteneurs et les distributions Kubernetes, y compris Red Hat OpenShift. Pour en savoir plus, rendez-vous sur le site Web Astra Trident ["ici"](#).

Les pages suivantes présentent des informations supplémentaires sur les produits NetApp validés pour la gestion du stockage persistant et des applications dans la solution Red Hat OpenShift avec NetApp :

- "NetApp Astra Control Center"
- "NetApp Astra Trident"

Présentation de NetApp Astra Control Center

NetApp Astra Control Center propose un ensemble complet de services de gestion du stockage et des données respectueuse des applications pour les workloads Kubernetes avec état, déployés dans un environnement sur site et optimisé par les technologies NetApp de protection des données.



NetApp Astra Control Center peut être installé sur un cluster Red Hat OpenShift que l'orchestrateur de stockage Astra Trident est déployé et configuré avec des classes de stockage et des systèmes back-end de stockage dans des systèmes de stockage NetApp ONTAP.

Pour l'installation et la configuration d'Astra Trident pour prendre en charge Astra Control Center, voir ["ce document ici"](#).

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 contrôle limité et une télémétrie (valeurs de metrics de 7 jours) sont disponibles et exportés vers les outils de contrôle natifs Kubernetes (Prometheus et Grafana) via des terminaux ouverts.

Le centre de contrôle Astra est entièrement intégré à l'écosystème NetApp AutoSupport et Active IQ qui fournit un soutien aux utilisateurs, fournit des conseils pour la résolution de problèmes et affiche des statistiques d'utilisation.

En plus de la version payante d'Astra Control Center, une licence d'évaluation de 90 jours est disponible. La

version d'évaluation est prise en charge par e-mail et par la communauté (Channel Slack). Les clients ont accès à ces articles, ainsi qu'à d'autres articles de la base de connaissances, et à la documentation disponible dans le tableau de bord de support des produits.

Pour commencer avec NetApp Astra Control Center, rendez-vous sur le ["Site Web d'Astra"](#).

Conditions préalables à l'installation d'Astra Control Center

1. Un ou plusieurs clusters Red Hat OpenShift. Les versions 4.6 EUS et 4.7 sont actuellement prises en charge.
2. Astra Trident doit déjà être installé et configuré sur chaque cluster Red Hat OpenShift.
3. Un ou plusieurs systèmes de stockage NetApp ONTAP exécutant ONTAP 9.5 ou version ultérieure.



Il est recommandé que chaque installation OpenShift sur un site dispose d'un SVM dédié pour le stockage persistant. Les déploiements multisites requièrent des systèmes de stockage supplémentaires.

4. Un système back-end de stockage Trident doit être configuré sur chaque cluster OpenShift avec un SVM sauvegardé par un cluster ONTAP.
5. Classe de stockage par défaut configurée sur chaque cluster OpenShift avec Astra Trident comme provisionneur de stockage.
6. Un équilibreur de charge doit être installé et configuré sur chaque cluster OpenShift pour équilibrer les charges et exposer les services OpenShift.



Voir le lien ["ici"](#) pour plus d'informations sur les équilibreurs de charge qui ont été validés à cet effet.

7. Un registre d'images privées doit être configuré pour héberger les images du NetApp Astra Control Center.



Voir le lien ["ici"](#) Pour installer et configurer un registre privé OpenShift à cet effet.

8. Vous devez disposer d'un accès Cluster Admin au cluster Red Hat OpenShift.
9. Vous devez disposer d'un accès d'administration aux clusters NetApp ONTAP.
10. Une station de travail d'administration avec docker ou podman, tridentctl et oc ou kubectl a été installée et ajoutée à votre \$PATH



Les installations Docker doivent avoir une version docker supérieure à 20.10 et les installations Podman doivent avoir une version podman supérieure à 3.0.

Poser le centre de contrôle Astra

Utilisation de OperatorHub

1. Connectez-vous au site de support NetApp et téléchargez la dernière version de NetApp Astra Control Center. Une licence associée à votre compte NetApp est requise. Après avoir téléchargé le fichier tarball, transférez-le sur le poste de travail d'administration.



Pour commencer avec une licence d'essai d'Astra Control, visitez le ["Site d'inscription à Astra"](#).

2. Déballez la boule tar et remplacez le répertoire de travail par le dossier obtenu.

```
[netapp-user@rhel7 ~]$ tar -vxzf astra-control-center-21.12.60.tar.gz
[netapp-user@rhel7 ~]$ cd astra-control-center-21.12.60
```

3. Avant de commencer l'installation, poussez les images du centre de contrôle Astra vers un registre d'images. Vous pouvez choisir de le faire avec Docker ou Podman, les instructions pour les deux sont fournies dans cette étape.

Podman

- a. Exportez le FQDN du Registre avec le nom de l'organisation/espace de noms/projet comme variable d'environnement 'regiant'.

```
[netapp-user@rhel7 ~]$ export REGISTRY=astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra
```

- b. Connectez-vous au registre.

```
[netapp-user@rhel7 ~]$ podman login -u ocp-user -p password --tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com
```



Si vous utilisez kubeadmin l'utilisateur doit se connecter au registre privé, puis utiliser un jeton au lieu du mot de passe - `podman login -u ocp-user -p token --tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com`.



Vous pouvez également créer un compte de service, attribuer un rôle d'éditeur de registre et/ou de visualiseur de registre (selon que vous avez besoin d'un accès Push/Pull) et vous connecter au registre à l'aide du jeton du compte de service.

- c. Créez un fichier de script shell et collez le contenu suivant dans celui-ci.

```
[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar) ; do
    # Load to local cache. And store the name of the loaded
    image trimming the 'Loaded images: '
    astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image(s): //' )
    astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
    # Tag with local image repo.
    podman tag ${astraImage} ${REGISTRY}/${astraImage}
    # Push to the local repo.
    podman push ${REGISTRY}/${astraImage}
done
```



Si vous utilisez des certificats non approuvés pour votre registre, modifiez le script de shell et utilisez-le `--tls-verify=false` pour la commande `push` `podman podman push $REGISTRY/$(echo $astraImage | sed 's/[/\]\+\\///') --tls-verify=false`.

d. Rendre le fichier exécutable.

```
[netapp-user@rhel7 ~]$ chmod +x push-images-to-registry.sh
```

e. Exécutez le script de shell.

```
[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
```


Docker

- a. Exportez le FQDN du Registre avec le nom de l'organisation/espace de noms/projet comme variable d'environnement 'registry'.

```
[netapp-user@rhel7 ~]$ export REGISTRY=astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra
```

- b. Connectez-vous au registre.

```
[netapp-user@rhel7 ~]$ docker login -u ocp-user -p password astra-registry.apps.ocp-vmw.cie.netapp.com
```



Si vous utilisez kubeadmin l'utilisateur doit se connecter au registre privé, puis utiliser un jeton au lieu du mot de passe - `docker login -u ocp-user -p token astra-registry.apps.ocp-vmw.cie.netapp.com`.



Vous pouvez également créer un compte de service, attribuer un rôle d'éditeur de registre et/ou de visualiseur de registre (selon que vous avez besoin d'un accès Push/Pull) et vous connecter au registre à l'aide du jeton du compte de service.

- c. Créez un fichier de script shell et collez le contenu suivant dans celui-ci.

```
[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar) ; do
    # Load to local cache. And store the name of the loaded
    image trimming the 'Loaded images: '
    astraImage=$(docker load --input ${astraImageFile} | sed
's/Loaded image: //' )
    astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
    # Tag with local image repo.
    docker tag ${astraImage} ${REGISTRY}/${astraImage}
    # Push to the local repo.
    docker push ${REGISTRY}/${astraImage}
done
```

- d. Rendre le fichier exécutable.

```
[netapp-user@rhel7 ~]$ chmod +x push-images-to-registry.sh
```

- e. Exécutez le script de shell.

```
[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
```

4. Lorsque vous utilisez des registres d'images privés qui ne sont pas de confiance publique, chargez les certificats TLS du registre d'images sur les nœuds OpenShift. Pour ce faire, créez une config map dans l'espace de noms openshift-config à l'aide des certificats TLS et installez-la sur la configuration d'images du cluster pour que le certificat soit fiable.

```
[netapp-user@rhel7 ~]$ oc create configmap default-ingress-ca -n  
openshift-config --from-file=astra-registry.apps.ocp  
-vmw.cie.netapp.com=tls.crt  
  
[netapp-user@rhel7 ~]$ oc patch image.config.openshift.io/cluster  
--patch '{"spec":{"additionalTrustedCA":{"name":"default-ingress-  
ca"}}}' --type=merge
```



Si vous utilisez un registre interne OpenShift avec des certificats TLS par défaut de l'opérateur d'entrée portant une route, vous devez suivre l'étape précédente pour corriger le nom d'hôte de la route. Pour extraire les certificats de l'opérateur Ingress, vous pouvez utiliser la commande `oc extract secret/router-ca --keys=tls.crt -n openshift-ingress-operator`.

5. Créer un espace de noms netapp-acc-operator Pour Astra Control Center.

```
[netapp-user@rhel7 ~]$ oc create ns netapp-acc-operator  
  
namespace/netapp-acc-operator created
```


6. Créez un secret avec des informations d'identification pour vous connecter au registre d'images dans netapp-acc-operator espace de noms.

```
[netapp-user@rhel7 ~]$ oc create secret docker-registry astra-  
registry-cred --docker-server=astra-registry.apps.ocp  
-vmw.cie.netapp.com --docker-username=ocp-user --docker  
-password=password -n netapp-acc-operator  
  
secret/astra-registry-cred created
```

7. Connectez-vous à la console IUG de Red Hat OpenShift avec un accès cluster-admin.
8. Sélectionnez Administrateur dans la liste déroulante perspective.
9. Accédez à Operators > OperatorHub et recherchez Astra.



10. Sélectionnez `netapp-acc-operator` mosaïque et clic `Install`.



netapp-acc-operator
21.12.63-1 provided by NetApp
✕

Install

Latest version 21.12.63-1	Astra Control is an application-aware data management solution that manages, protects and moves data-rich Kubernetes workloads in both public clouds and on-premises.
Capability level <input checked="" type="radio"/> Basic Install <input type="radio"/> Seamless Upgrades <input type="radio"/> Full Lifecycle <input type="radio"/> Deep Insights <input type="radio"/> Auto Pilot	Astra Control enables data protection, disaster recovery, and migration for your Kubernetes workloads, leveraging NetApp's industry-leading data management technology for snapshots, backups, replication and cloning.
Provider type Certified	How to deploy Astra Control Refer to Installation Procedure to deploy Astra Control Center using the Operator.
Provider NetApp	Documentation Refer to Astra Control Center Documentation to complete the setup and start managing applications.

11. Sur l'écran installer l'opérateur, acceptez tous les paramètres par défaut et cliquez sur `Install`.

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel *

- ☐ alpha
- ☒ stable

Installation mode *

- ☒ All namespaces on the cluster (default)
Operator will be available in all Namespaces.
- ☐ A specific namespace on the cluster
This mode is not supported by this Operator

Installed Namespace *

PR netapp-acc-operator (Operator recommended)

⚠ Namespace already exists

Namespace **netapp-acc-operator** already exists and will be used. Other users can already have access to this namespace.

Approval strategy *

- ☒ Automatic
- ☐ Manual

Install

Cancel

 **netapp-acc-operator**
provided by NetApp

Provided APIs

 **Astra Control Center**

AstraControlCenter is the Schema for the astracontrolcenters API

12. Attendre la fin de l'installation par l'opérateur.



netapp-acc-operator
21.12.63-1 provided by NetApp



Installing Operator

InstallWaiting: installing; waiting for deployment acc-operator-controller-manager to become ready: Waiting for rollout to finish: 0 of 1 updated replicas are available...

The Operator is being installed. This may take a few minutes.

[View installed Operators in Namespace netapp-acc-operator](#)

13. Une fois l'installation de l'opérateur réussie, cliquez sur View Operator.



netapp-acc-operator
21.12.63-1 provided by NetApp



Installed operator - ready for use

[View Operator](#)[View installed Operators in Namespace netapp-acc-operator](#)

14. Cliquez ensuite sur `Create Instance` Dans la mosaïque Astra Control Center du conducteur.

[Installed Operators](#) > [Operator details](#)



netapp-acc-operator
21.12.63-1 provided by NetApp

[Details](#)

[YAML](#)

[Subscription](#)

[Events](#)

[Astra Control Center](#)

Provided APIs

ACC Astra Control Center

AstraControlCenter is the Schema for
the astracontrolcenters API

[+ Create instance](#)

15. Remplissez le `Create AstraControlCenter` et cliquez sur `Create`.

- Vous pouvez modifier le nom de l'instance du Centre de contrôle Astra.
- Vous pouvez éventuellement activer ou désactiver Auto support. Il est recommandé de conserver la fonctionnalité Auto support.
- Saisissez le nom de domaine complet pour Astra Control Center.
- Accédez à la version du Centre de contrôle Astra ; la dernière est affichée par défaut.

- e. Entrez un nom de compte pour le centre de contrôle Astra et des détails d'administrateur tels que le prénom, le nom et l'adresse e-mail.
- f. Entrez la règle de récupération du volume. La valeur par défaut est conservation.
- g. Dans le Registre d'images, entrez le FQDN de votre registre ainsi que le nom d'organisation tel qu'il a été donné lors de l'envoi des images au Registre (dans cet exemple, astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra)
- h. Si vous utilisez un registre qui nécessite une authentification, entrez le nom secret dans la section Registre d'images.
- i. Configurez les options d'échelle pour les limites de ressources Astra Control Center.
- j. Entrez le nom de la classe de stockage si vous souhaitez placer des ESV sur une classe de stockage non-défaut.
- k. Définissez les préférences de gestion de CRD.

Project: netapp-acc-operator ▼

Name *

astra

Labels

app=frontend

Account Name *

HCG Solutions Engineering

Astra Control Center account name

Astra Address *

astra-control-center.cie.netapp.com

AstraAddress defines how Astra will be found in the data center. This IP address and/or DNS A record must be created prior to provisioning Astra Control Center. Example - "astra.example.com" The A record and its IP address must be allocated prior to provisioning Astra Control Center

Astra Version *

21.12.60

Version of AstraControlCenter to deploy. You are provided a Helm repository with a corresponding version. Example - 1.5.2, 1.4.2-patch

Email *

solutions_tme@netapp.com

EmailAddress will be notified by Astra as events warrant.

Auto Support *



AutoSupport indicates willingness to participate in NetApp's proactive support application, NetApp Active IQ. The default election is true and indicates support data will be sent to NetApp. An empty or blank election is the same as a default election. Air gapped installations should enter false.

First Name

HCG

The first name of the SRE supporting Astra.

Last Name

Admin

The last name of the SRE supporting Astra.

Image Registry

The container image registry that is hosting the Astra application images, ACC Operator and ACC Helm Repository.

Name

astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra

The name of the image registry. For example "example.registry/astra". Do not prefix with protocol.

Secret

astra-registry-cred

The name of the Kubernetes secret that will authenticate with the image registry.

Volume Reclaim Policy

Retain

Reclaim policy to be set for persistent volumes

Astra Resources Scaler

Default

Scaling options for AstraControlCenter Resource limits.

Storage Class

The storage class to be used for PVCs. If not set, default storage class will be used.

Crds

Options for how ACC should handle CRDs.

Create

Cancel

Automatisation [Ansible]

1. Pour déployer Astra Control Center sur un playbooks Ansible, vous devez utiliser un ordinateur Ubuntu/RHEL avec Ansible installé. Suivre les procédures "[ici](#)" Pour Ubuntu et RHEL.
2. Clonez le référentiel GitHub qui héberge le contenu Ansible.

```
git clone https://github.com/NetApp-
Automation/na_astra_control_suite.git
```

3. Connectez-vous au site de support NetApp et téléchargez la dernière version de NetApp Astra Control Center. Une licence associée à votre compte NetApp est requise. Après avoir téléchargé le tarball, transférez-le sur le poste de travail.



Pour commencer avec une licence d'essai d'Astra Control, visitez le "[Site d'inscription à Astra](#)".

4. Créez ou obtenez le fichier kubeconfig avec un accès administrateur au cluster OpenShift sur lequel vous devez installer Astra Control Center.

5. Remplacez le répertoire par `na_astra_control_suite`.

```
cd na_astra_control_suite
```

6. Modifiez le `vars/vars.yml` et remplissez les variables avec les informations requises.

```
#Define whether or not to push the Astra Control Center images to
your private registry [Allowed values: yes, no]
push_images: yes

#The directory hosting the Astra Control Center installer
installer_directory: /home/admin/

#Specify the ingress type. Allowed values - "AccTraefik" or
"Generic"
#"AccTraefik" if you want the installer to create a LoadBalancer
type service to access ACC, requires MetallB or similar.
#"Generic" if you want to create or configure ingress controller
yourself, installer just creates a ClusterIP service for traefik.
ingress_type: "AccTraefik"

#Name of the Astra Control Center installer (Do not include the
extension, just the name)
astra_tar_ball_name: astra-control-center-22.04.0

#The complete path to the kubeconfig file of the
kubernetes/openshift cluster Astra Control Center needs to be
installed to.
hosting_k8s_cluster_kubeconfig_path: /home/admin/cluster-
kubeconfig.yml

#Namespace in which Astra Control Center is to be installed
astra_namespace: netapp-astra-cc

#Astra Control Center Resources Scaler. Leave it blank if you want
to accept the Default setting.
astra_resources_scaler: Default

#Storageclass to be used for Astra Control Center PVCs, it must be
created before running the playbook [Leave it blank if you want the
PVCs to use default storageclass]
astra_trident_storageclass: basic

#Reclaim Policy for Astra Control Center Persistent Volumes [Allowed
values: Retain, Delete]
```



```

storageclass_reclaim_policy: Retain

#Private Registry Details
astra_registry_name: "docker.io"

#Whether the private registry requires credentials [Allowed values:
yes, no]
require_reg_creds: yes

#If require_reg_creds is yes, then define the container image
registry credentials
#Usually, the registry namespace and usernames are same for
individual users
astra_registry_namespace: "registry-user"
astra_registry_username: "registry-user"
astra_registry_password: "password"

#Kubernetes/OpenShift secret name for Astra Control Center
#This name will be assigned to the K8s secret created by the
playbook
astra_registry_secret_name: "astra-registry-credentials"

#Astra Control Center FQDN
acc_fqdn_address: astra-control-center.cie.netapp.com

#Name of the Astra Control Center instance
acc_account_name: ACC Account Name

#Administrator details for Astra Control Center
admin_email_address: admin@example.com
admin_first_name: Admin
admin_last_name: Admin

```

7. Utilisez le PlayBook pour déployer le centre de contrôle Astra. Le PlayBook requiert des privilèges root pour certaines configurations.

Si l'utilisateur exécutant le PlayBook est root ou a configuré un sudo sans mot de passe, exécutez la commande suivante pour exécuter le PlayBook.

```
ansible-playbook install_acc_playbook.yml
```

Si l'accès sudo basé sur un mot de passe est configuré, exécutez la commande suivante pour exécuter le PlayBook, puis saisissez le mot de passe sudo.

```
ansible-playbook install_acc_playbook.yml -K
```

Après l'installation

1. L'installation peut prendre plusieurs minutes. Vérifier que tous les pods et services dans le `netapp-astra-cc` les espaces de noms sont opérationnels.

```
[netapp-user@rhel7 ~]$ oc get all -n netapp-astra-cc
```

2. Vérifier le `acc-operator-controller-manager` journaux pour vérifier que l'installation est terminée.

```
[netapp-user@rhel7 ~]$ oc logs deploy/acc-operator-controller-manager -n netapp-acc-operator -c manager -f
```



Le message suivant indique que le centre de contrôle Astra a été installé avec succès.

```
{"level":"info","ts":1624054318.029971,"logger":"controllers.AstraControlCenter","msg":"Successfully Reconciled AstraControlCenter in [seconds]s","AstraControlCenter":"netapp-astra-cc/astra","ae.Version":"[21.12.60]"}

```

3. Le nom d'utilisateur pour la connexion à Astra Control Center est l'adresse électronique de l'administrateur fournie dans le fichier CRD et le mot de passe est une chaîne ACC- Joint à l'UUID du centre de contrôle Astra. Exécutez la commande suivante :

```
[netapp-user@rhel7 ~]$ oc get astracontrolcenters -n netapp-astra-cc
NAME      UUID
astra     345c55a5-bf2e-21f0-84b8-b6f2bce5e95f
```



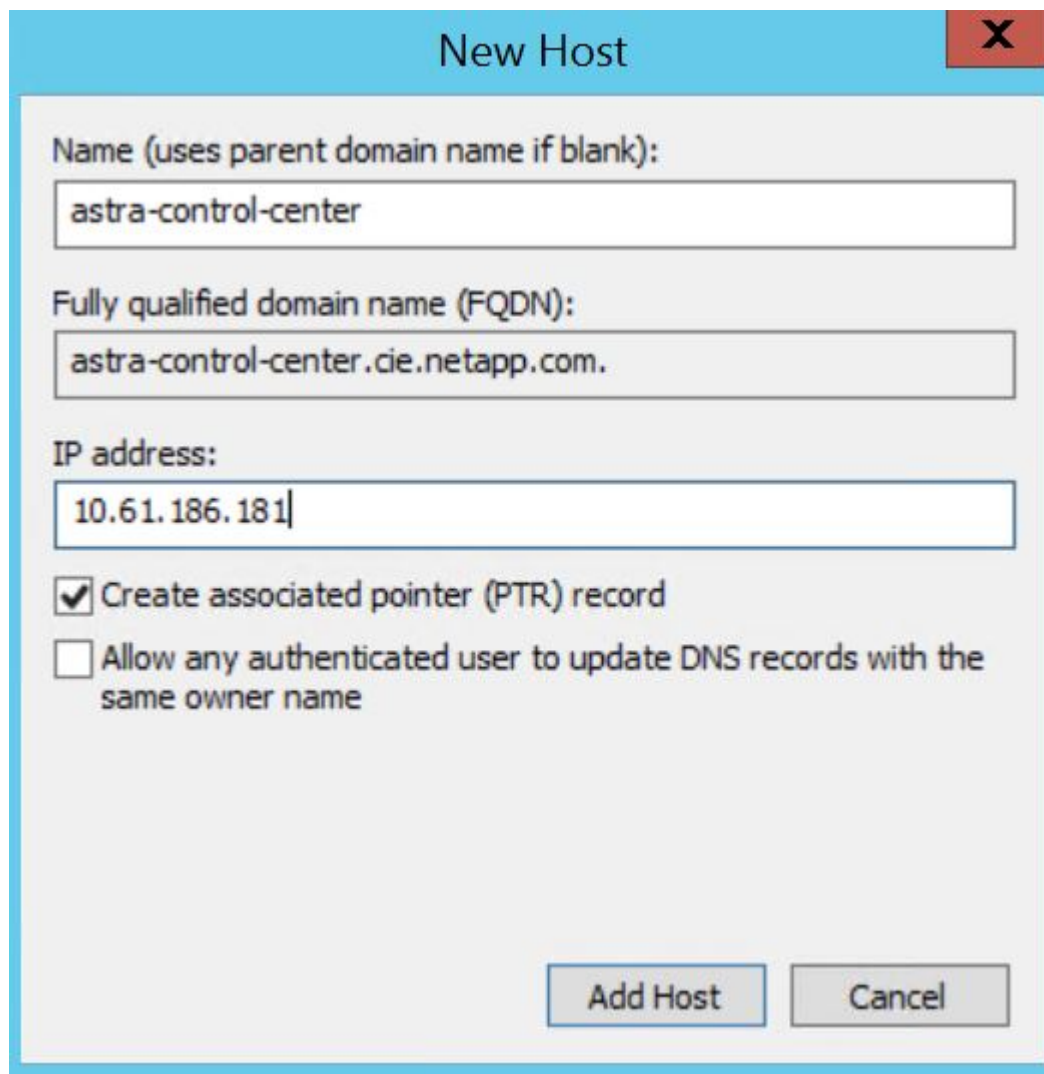
Dans cet exemple, le mot de passe est ACC-345c55a5-bf2e-21f0-84b8-b6f2bce5e95f.

4. Procurez-vous l'IP d'équilibrage de charge du service traefik.

```
[netapp-user@rhel7 ~]$ oc get svc -n netapp-astra-cc | egrep 'EXTERNAL|traefik'
```

NAME	TYPE	CLUSTER-IP
EXTERNAL-IP	PORT(S)	
AGE		
traefik	LoadBalancer	172.30.99.142
10.61.186.181	80:30343/TCP, 443:30060/TCP	
16m		

5. Ajoutez une entrée dans le serveur DNS pointant le FQDN fourni dans le fichier CRD Astra Control Center vers le `EXTERNAL-IP` du service de trafik.



The image shows a 'New Host' dialog box with a light blue header and a red close button in the top right corner. The dialog contains three text input fields and two checkboxes. The first field, labeled 'Name (uses parent domain name if blank):', contains the text 'astra-control-center'. The second field, labeled 'Fully qualified domain name (FQDN):', contains 'astra-control-center.cie.netapp.com.'. The third field, labeled 'IP address:', contains '10.61.186.181'. Below the fields are two checkboxes: the first is checked and labeled 'Create associated pointer (PTR) record', and the second is unchecked and labeled 'Allow any authenticated user to update DNS records with the same owner name'. At the bottom right are two buttons: 'Add Host' and 'Cancel'.

New Host

Name (uses parent domain name if blank):
astra-control-center

Fully qualified domain name (FQDN):
astra-control-center.cie.netapp.com.

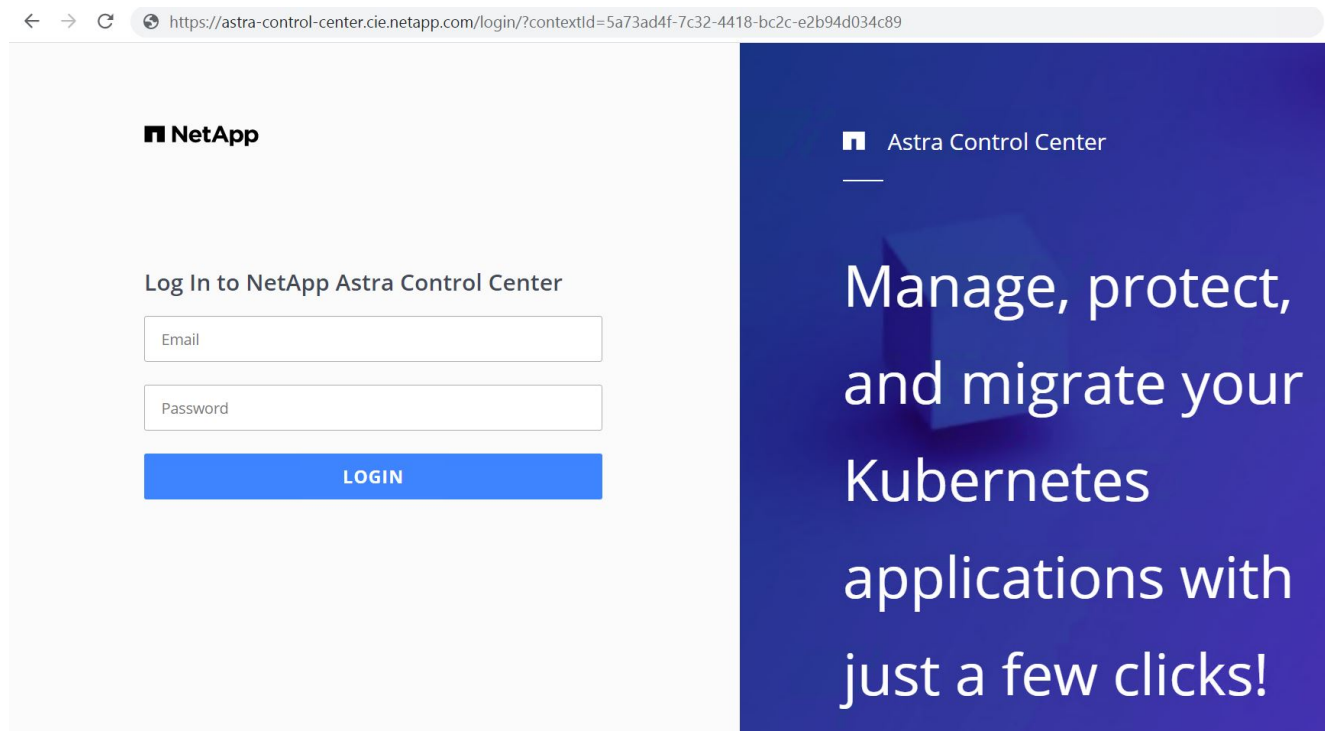
IP address:
10.61.186.181

☒ Create associated pointer (PTR) record

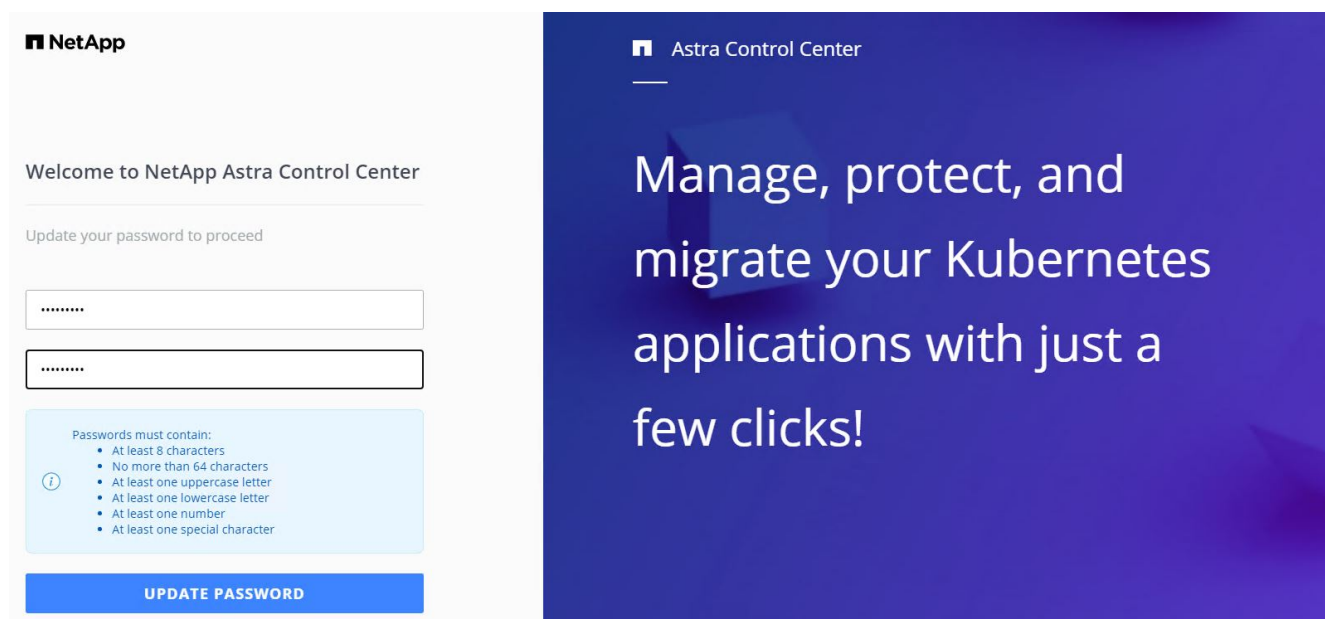
☐ Allow any authenticated user to update DNS records with the same owner name

Add Host Cancel

6. Connectez-vous à l'interface graphique d'Astra Control Center en parcourant son FQDN.



7. Lorsque vous vous connectez à l'interface graphique d'Astra Control Center pour la première fois à l'aide de l'adresse e-mail d'administration fournie dans CRD, vous devez modifier le mot de passe.



8. Si vous souhaitez ajouter un utilisateur au Centre de contrôle Astra, accédez à compte > utilisateurs, cliquez sur Ajouter, entrez les détails de l'utilisateur et cliquez sur Ajouter.

Add user
✕

USER DETAILS

First name

Nikhil

Last name

Kulkarni

Email address

tme_nik@netapp.com

PASSWORD

Temporary password

Confirm temporary password

ⓘ

Passwords must contain:

- At least 8 characters
- No more than 64 characters
- At least one lowercase letter
- At least one uppercase letter
- At least one number
- At least one special character

USER ROLE ⓘ

Role

Owner

▼

Cancel

Add ✓

ADD NEW USER

Add new user

Add a new user to your Astra Control Center account. New users will be prompted to update their password the first time they log in to Astra Control Center. They will also inherit access to account-wide credentials according to their role. Read more in [users](#) .

- Astra Control Center requiert une licence pour toutes ses fonctionnalités. Pour ajouter une licence, accédez à compte > Licence, cliquez sur Ajouter une licence et téléchargez le fichier de licence.

Account

Users

Credentials

Notifications

License

Connections

ASTRA CONTROL CENTER LICENSE

To get started with Astra Control Center, select Add license to manually upload the file.

ADD LICENSE

Select and add a license file.

License file

EvalNLF-AstraControlCenter-480Cores(vCPU)-100000002-ACC60f19...

⬆

✕

Cancel

Add

En cas de problème avec l'installation ou la configuration de NetApp Astra Control Center, la base de connaissances des problèmes connus est disponible ["ici"](#).


Enregistrez vos clusters Red Hat OpenShift avec Astra Control Center

Pour permettre à Astra Control Center de gérer vos charges de travail, vous devez d'abord enregistrer votre cluster Red Hat OpenShift.


19

Enregistrez les clusters Red Hat OpenShift

1. La première étape consiste à ajouter les clusters OpenShift au Centre de contrôle Astra et à les gérer. Accédez aux clusters et cliquez sur Ajouter un cluster, téléchargez le fichier kubeconfig pour le cluster OpenShift, puis cliquez sur Sélectionner un stockage.

 **Add cluster**

STEP 1/3: CREDENTIALS





CREDENTIALS

Provide Astra Control access to your Kubernetes and OpenShift clusters by entering a kubeconfig credential.


Follow [instructions](#) on how to create a dedicated admin-role kubeconfig.

[Upload file](#) | Paste from clipboard

Kubeconfig YAML file
ocp-vmw kubeconfig.txt


 

Credential name
ocp-vmw

 **ADDING A CLUSTER**

Adding a cluster is needed for Astra Control to discover your Kubernetes applications.

Select a cloud provider and input credentials to get started.

Read more in [Clusters](#) .

Cancel

Configure storage →



Le fichier kubeconfig peut être généré pour s'authentifier avec un nom d'utilisateur et un mot de passe ou un jeton. Les jetons expirent après un délai limité et peuvent laisser le cluster enregistré inaccessible. NetApp recommande d'utiliser un fichier kubeconfig avec un nom d'utilisateur et un mot de passe pour enregistrer vos clusters OpenShift sur Astra Control Center.

2. Astra Control Center détecte les classes de stockage admissibles. Maintenant, sélectionnez la façon dont storageclass provisionne les volumes en utilisant Trident sauvegardé par un SVM sur NetApp ONTAP et Click Review. Dans le volet suivant, vérifiez les détails et cliquez sur Ajouter un cluster.

STORAGE

Existing storage classes are discovered and verified as eligible for use with Astra Control. You can use your existing default, or choose to set a new default at this time.
Applications with persistent volumes on eligible storage classes are validated for use with Astra Control.

Set default	Storage class	Storage provisioner	Reclaim policy	Binding mode	Eligible
<input checked="" type="radio"/>	ocp-trident <small>Default</small>	csi.trident.netapp.io	Delete	Immediate	
<input type="radio"/>	ocp-trident-iscsi	csi.trident.netapp.io	Delete	Immediate	
<input type="radio"/>	project-1-sc	csi.trident.netapp.io	Delete	Immediate	
<input type="radio"/>	thin	kubernetes.io/vsphere-volume	Delete	Immediate	

[← Select credentials](#)
[Review →](#)

- Enregistrez les deux clusters OpenShift comme décrit à l'étape 1. Lorsqu'elles sont ajoutées, les clusters passent à l'état découverte pendant qu'Astra Control Center les inspecte et installe les agents nécessaires. L'état du cluster est modifié en cours d'exécution après son enregistrement.

admin

Dashboard

MANAGE YOUR APPS

Apps

Clusters

MANAGE YOUR STORAGE

Backends

Buckets

MANAGE YOUR ACCOUNT

Account

Activity

Support

Clusters

Actions

+ Add

Search

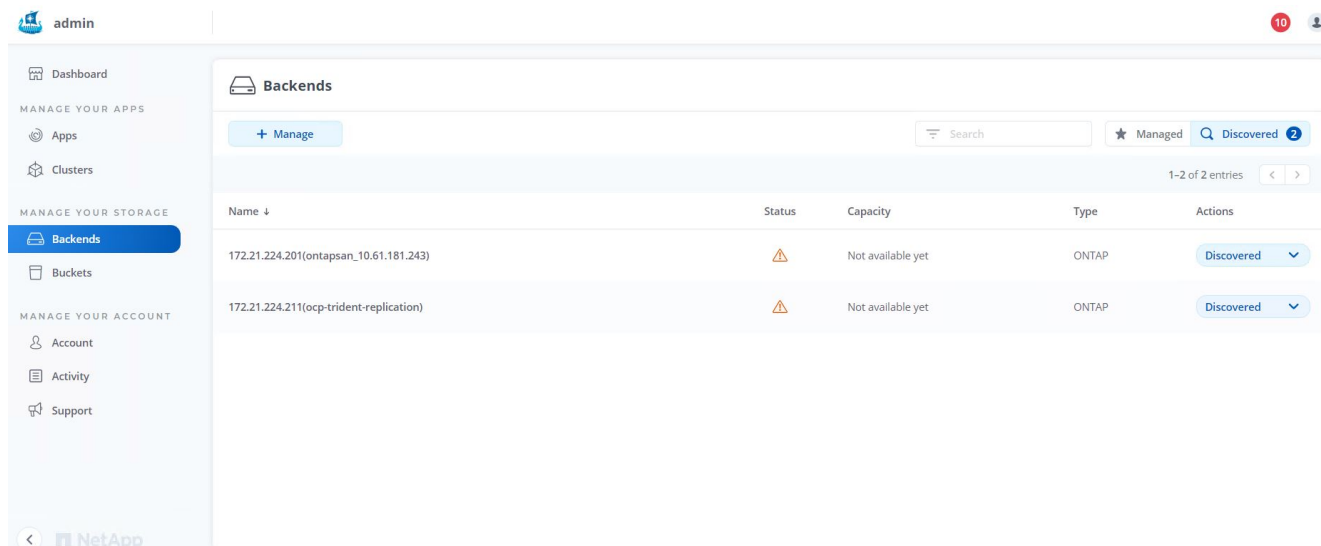
1-2 of 2 entries

<input type="checkbox"/>	Name ↓	Ready	Type	Version	Actions
<input type="checkbox"/>	ocp-vmw		Red Hat OpenShift	v1.20.0+df9c838	Running
<input type="checkbox"/>	ocp-vmware2		Red Hat OpenShift	v1.20.0+c8905da	Running



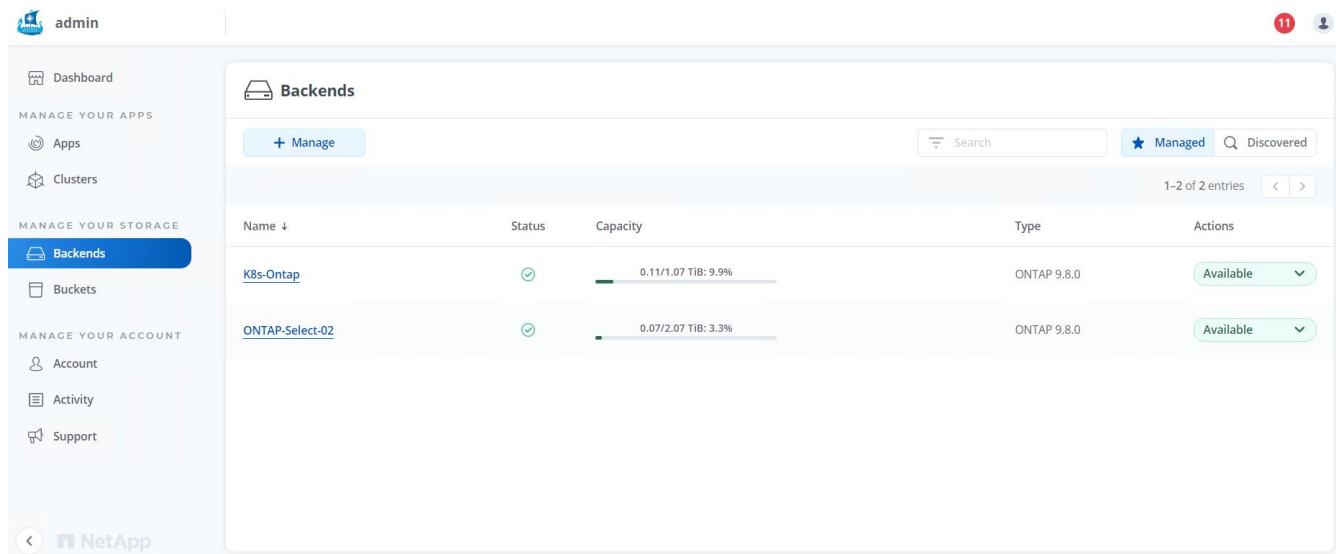
Tous les clusters Red Hat OpenShift devant être gérés par Astra Control Center doivent avoir accès au registre d'images utilisé pour son installation lorsque les agents installés sur les clusters gérés extraient les images de ce registre.

- Importation de clusters ONTAP comme ressources de stockage à gérer en tant que système back-end par Astra Control Center. Lorsque des clusters OpenShift sont ajoutés à Astra et qu'un storageclass est configuré, il détecte et inspecte automatiquement le cluster ONTAP qui soutient le storageclass, mais ne l'importe pas dans le Control Center Astra à gérer.



5. Pour importer les clusters ONTAP, accédez aux systèmes back-end, cliquez sur la liste déroulante et sélectionnez gérer en regard du cluster ONTAP à gérer. Entrez les informations d'identification du cluster ONTAP, cliquez sur vérifier les informations, puis sur Importer le stockage back-end.

6. Une fois que le système back-end est ajouté, le statut devient disponible. Ces systèmes back-end disposent désormais d'informations sur les volumes persistants dans le cluster OpenShift et sur les volumes correspondants sur le système ONTAP.



7. Pour la sauvegarde et la restauration entre les clusters OpenShift avec Astra Control Center, vous devez provisionner un compartiment de stockage objet qui prend en charge le protocole S3. Les options ONTAP S3, StorageGRID et AWS S3 sont actuellement prises en charge. Pour les besoins de cette installation, nous allons configurer un compartiment AWS S3. Accédez à godets, cliquez sur Ajouter un compartiment et sélectionnez Generic S3. Entrez les informations d'identification du compartiment S3 et des informations d'identification pour y accéder, cochez la case « définir ce compartiment par défaut pour le cloud », puis cliquez sur Ajouter.

Add bucket
×

STORAGE BUCKET

Enter the access details of your existing object store bucket to allow Astra Control to store your application backups.

Type

Generic S3

Existing bucket name

ocp-vmware2-astra-cc

Description (optional)

S3 server name or IP address

s3.us-east-1.amazonaws.com

☒ Make this bucket the default bucket for this cloud

SELECT CREDENTIALS

Astra Control requires S3 access credentials with the roles necessary to facilitate Kubernetes application data management.

Add

Use existing

Access ID

AMW5TCKDSU6HWSZXABD

Secret key

.....

Credential name

AWS-S3

Cancel

Add ✓

ADDING STORAGE BUCKETS

Astra Control stores backups in your existing object store buckets. The first bucket added for a selected cloud will be designated as the default bucket for backup and clone operations.

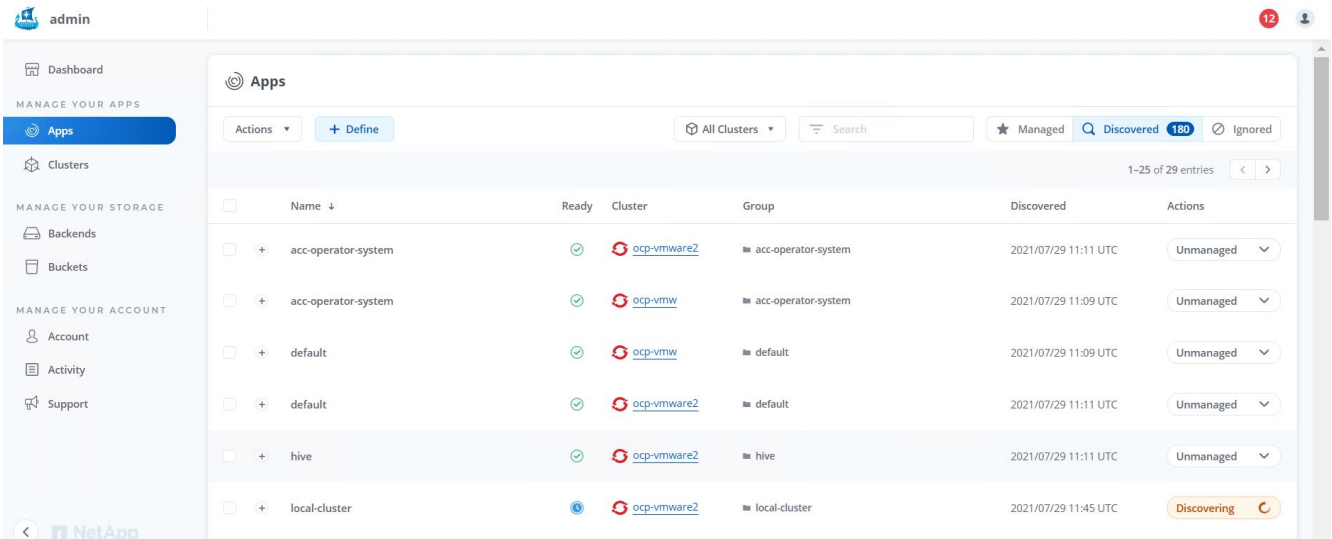
Read more in [storage buckets](#).

Choisissez les applications à protéger

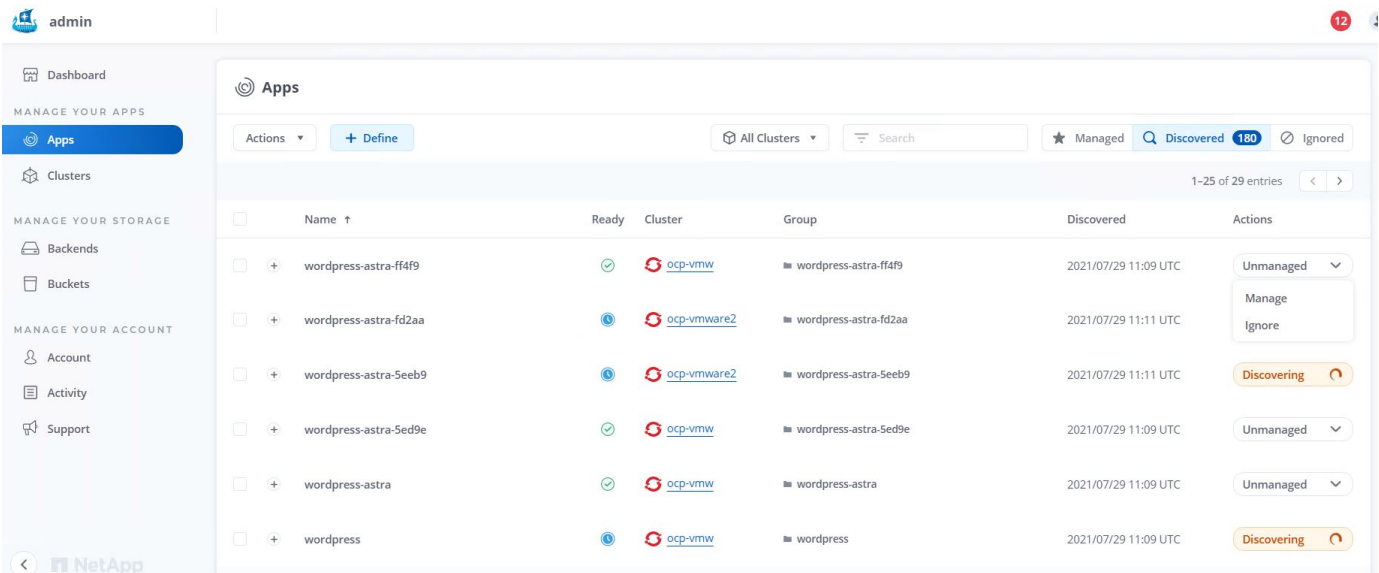
Une fois vos clusters Red Hat OpenShift enregistrés, vous pouvez détecter les applications déployées et les gérer via Astra Control Center.

Gestion des applications

1. Une fois les clusters OpenShift et les systèmes back-end ONTAP enregistrés auprès de l'Astra Control Center, le centre de contrôle démarre automatiquement la détection des applications dans tous les namespaces qui utilisent le storageclass configuré avec le back-end ONTAP spécifié.



2. Accédez à applications > découverte et cliquez sur le menu déroulant en regard de l'application que vous souhaitez gérer à l'aide d'Astra. Cliquez ensuite sur gérer.



1. L'application passe à l'état disponible et peut être affichée sous l'onglet géré de la section applications.

Apps

Actions ▾

+ Define

All Clusters ▾

⌵

Search

★ Managed


🔍 Discovered 175

🚫 Ignored

1-1 of 1 entries

<

>

<input type="checkbox"/>	Name ↓	Ready	Protected	Cluster	Group	Discovered	Actions
<input type="checkbox"/>	wordpress-astra-ff4f9	✔	ℹ	 ocp-vmw	■ wordpress-astra-ff4f9	2021/07/29 11:09 UTC	Available ▾

Protégez vos applications

Une fois les charges de travail applicatives gérées par Astra Control Center, vous pouvez configurer les paramètres de protection pour ces charges de travail.

Création d'un instantané d'application

Un snapshot d'une application crée une copie Snapshot ONTAP qui peut être utilisée pour restaurer ou cloner l'application à un point dans le temps spécifique en fonction de cette copie Snapshot.

1. Pour prendre un instantané de l'application, accédez à l'onglet applications > gestion, puis cliquez sur l'application dont vous souhaitez effectuer une copie Snapshot. Cliquez sur le menu déroulant en regard du nom de l'application et cliquez sur instantané.

wp

APPLICATION STATUS

✓ Healthy

APPLICATION PROTECTION STATUS

⚠ Unprotected

Images

docker.io/bitnami/mariadb:10.5.13-debian-10-r58

docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule

Disabled

Group

■ wp

Cluster

⚠

Running ▾

Snapshot

Backup

Clone

Restore

Unmanage

2. Entrez les détails du snapshot, cliquez sur Suivant, puis sur instantané. La création du Snapshot prend environ une minute et son état est disponible une fois celui-ci créé.

Snapshot application

STEP 1/2: DETAILS

X

SNAPSHOT DETAILS

Name

wp-snapshot-20220228185949

CREATING APPLICATION SNAPSHOTS

Astra Control can take a quick snapshot of your application configuration and persistent storage. Enter a snapshot name to get started.

Read more in [Protect apps](#).

Application

wp

Namespace

wp

Cluster

ocp-vmw

Cancel

Next →

Création d'une sauvegarde d'application

Une sauvegarde d'une application capture l'état actif de l'application et la configuration des ressources informatiques, les analyse en fichiers et les stocke dans un compartiment de stockage objet distant.

Pour la sauvegarde et la restauration des applications gérées dans le Centre de contrôle Astra, vous devez configurer les paramètres de superutilisateur des systèmes ONTAP de secours au préalable. Pour ce faire, entrez les commandes suivantes.

```
ONTAP::> export-policy rule modify -vserver ocp-trident -policyname
default -ruleindex 1 -superuser sys
ONTAP::> export-policy rule modify -policyname default -ruleindex 1 -anon
65534 -vserver ocp-trident
```

1. Pour créer une sauvegarde de l'application gérée dans Astra Control Center, accédez à l'onglet applications > géré et cliquez sur l'application dont vous souhaitez effectuer une sauvegarde. Cliquez sur le menu déroulant en regard du nom de l'application et cliquez sur Sauvegarder.

wp

APPLICATION STATUS

Healthy

APPLICATION PROTECTION STATUS

Unprotected

Images

docker.io/bitnami/mariadb:10.5.13-debian-10-r58

docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule

Disabled

Group

wp

Cluster

ocp-vmw

Running

Snapshot

Backup

Clone

Restore

Unmanage

2. Entrez les détails de la sauvegarde, sélectionnez le compartiment de stockage objet pour contenir les fichiers de sauvegarde, cliquez sur Next (Suivant) et, après avoir vérifié les détails, cliquez sur Backup (Sauvegarder). Selon la taille de l'application et des données, la sauvegarde peut prendre plusieurs minutes, et l'état de la sauvegarde est disponible une fois la sauvegarde terminée.

26

Backup application

STEP 1/2: DETAILS

X

BACKUP DETAILS

Name

wp-backup

☐ Backup from an existing snapshot

BACKUP DESTINATION

Bucket

na-ocp-astra/na-ocp-acc Available

CREATING APPLICATION BACKUPS

Astra Control can take a backup of your application configuration and persistent storage. Persistent storage backups are transferred to your object store. Enter a backup name to get started.

Read more in [Application backups](#).

Application

wp

Namespace

wp

Cluster

ocp-vmw

Cancel

Next →

Restauration d'une application

En appuyant sur un bouton, vous pouvez restaurer une application sur l'espace de noms d'origine dans le même cluster ou sur un cluster distant afin d'assurer la protection des applications et la reprise sur incident.

1. Pour restaurer une application, accédez à applications > onglet géré et cliquez sur l'application en question. Cliquez sur le menu déroulant en regard du nom de l'application et cliquez sur Restore.

wp

Running

APPLICATION STATUS

Healthy

APPLICATION PROTECTION STATUS

Partially protected

Images

docker.io/bitnami/mariadb:10.5.13-debian-10-r58

docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule

Disabled

Group

wp

Cluster

ocp-vmw

Snapshot

Backup

Clone

Restore

Unmanage

2. Entrez le nom de l'espace de noms de restauration, sélectionnez le cluster vers lequel vous souhaitez le restaurer et choisissez si vous souhaitez le restaurer à partir d'un snapshot existant ou à partir d'une sauvegarde de l'application. Cliquez sur Suivant.

Restore application

STEP 1/2: DETAILS

RESTORE DETAILS

Destination cluster

ocp-vmw

Destination namespace

wp

RESTORE SOURCE

Filter

Snapshots

Backups

Application backup	Ready	On-Schedule/On-Demand	Created ↑
wp-backup	✓	On-Demand	2022/02/28 18:54 UTC

RESTORING APPLICATIONS

Astra Control can restore your application configuration and persistent storage. Select a source snapshot or backup for the restored application.

- Application wp
- Namespace wp
- Cluster ocp-vmw

Cancel

Next →

- Dans le volet de révision, entrez `restore` Puis cliquez sur Restaurer une fois que vous avez examiné les détails.

Restore application

STEP 2/2: SUMMARY

REVIEW RESTORE INFORMATION

⚠

All existing resources associated with this application will be deleted and replaced with the source backup "wp-backup" taken on 2022/02/28 18:54 UTC. Persistent volumes will be deleted and recreated. External resources with dependencies on this application may be impacted.

We recommend taking a snapshot or a backup of your application before proceeding.

BACKUP

wp-backup

ORIGINAL GROUP

wp

ORIGINAL CLUSTER

ocp-vmw

RESOURCE LABELS

ClusterRole

kubernetes.io/bootstrapping: rbac-defaults +1

ClusterRoleBinding

RESTORE

wp

DESTINATION GROUP

wp

DESTINATION CLUSTER

ocp-vmw

RESOURCE LABELS

ClusterRole

kubernetes.io/bootstrapping: rbac-defaults +1

ClusterRoleBinding

Are you sure you want to restore the application "wp"?

Type **restore** below to confirm.

Confirm to restore

restore

← Back

Restore ✓

- La nouvelle application passe à l'état de restauration tandis qu'Astra Control Center restaure l'application sur le cluster sélectionné. Une fois que toutes les ressources de l'application sont installées et détectées par Astra, l'application passe à l'état disponible.

Actions

+ Define

Search

★

🔍

110

🚫

🔄

1-1 of 1 entries

<

>

<input type="checkbox"/>	Name ↓	Ready	Protected	Cluster	Group	Discovered	Actions
<input type="checkbox"/>	wp	✓	i	ocp-vmw	wp	2022/02/28 18:34 UTC	Available ▼

Clonage d'une application

Vous pouvez cloner une application sur le cluster d'origine ou sur un cluster distant à des fins de développement/test ou de protection des applications et de reprise sur incident. Le clonage d'une application au sein d'un même cluster sur le même système back-end utilise la technologie NetApp FlexClone, qui clonez instantanément les demandes de volume persistant et économise de l'espace de stockage.

1. Pour cloner une application, accédez à l'onglet applications > gestion et cliquez sur l'application en question. Cliquez sur le menu déroulant en regard du nom de l'application, puis cliquez sur Cloner.

wp

Running ▾

Snapshot
Backup
Clone
Restore
Unmanage

APPLICATION STATUS
Healthy

APPLICATION PROTECTION STATUS
Partially protected

Images
docker.io/bitnami/mariadb:10.5.13-debian-10-r58
docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule
Disabled

Group
wp

Cluster
ocp-vmw

2. Entrez les détails du nouveau namespace, sélectionnez le cluster vers lequel vous souhaitez le cloner à partir d'un snapshot existant ou d'une sauvegarde ou de l'état actuel de l'application. Cliquez ensuite sur Suivant et sur Cloner dans le volet d'évaluation une fois que vous avez passé en revue les détails.

Clone application

STEP 1/2: DETAILS

✕

CLONE DETAILS

Clone name
wp-clone

Clone namespace
wp-clone

Destination cluster
ocp-vmw

☐ Clone from an existing snapshot or backup

CLONING APPLICATIONS

Astra Control can create a clone of your application configuration and persistent storage. Persistent storage backups are transferred from your object store, so choosing a clone from an existing backup will complete the fastest. Enter a clone name to get started.

Read more in [Clone applications](#)

Application

wp

Namespace

wp


Cluster





ocp-vmw

Cancel







Next →

3. La nouvelle application passe à l'état découverte tandis que Astra Control Center crée l'application sur le cluster sélectionné. Une fois que toutes les ressources de l'application sont installées et détectées par Astra, l'application passe à l'état disponible.

 **Applications**

Actions ▾ [+ Define](#)  Search   110 

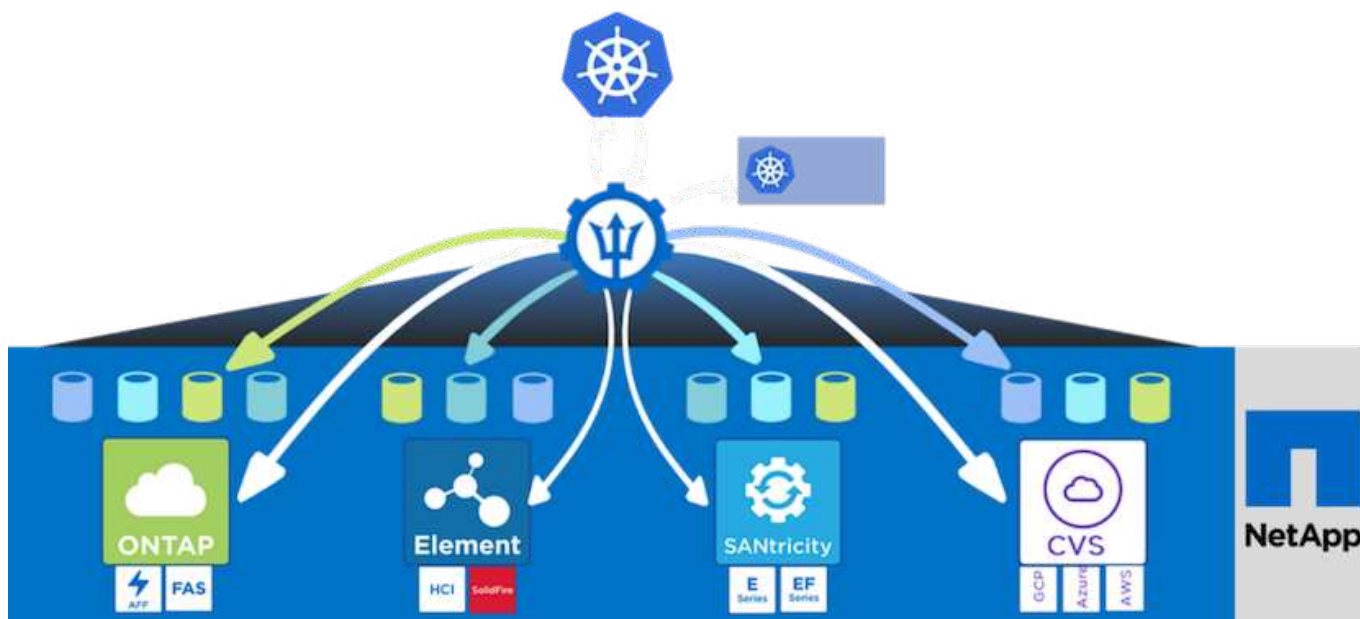
1-2 of 2 entries < >

<input type="checkbox"/>	Name ↓	Ready	Protected	Cluster	Group	Discovered	Actions
<input type="checkbox"/>	wp			 ocp-vmw	wp	2022/02/28 18:34 UTC	Available ▾
<input type="checkbox"/>	wp-clone			 ocp-vmw	wp-clone	2022/02/28 19:21 UTC	Available ▾

Présentation d'Astra Trident

Astra Trident est un orchestrateur de stockage open source entièrement pris en charge, y compris Red Hat OpenShift, pour les conteneurs et les distributions Kubernetes. Trident fonctionne avec l'ensemble de la gamme de solutions de stockage NetApp, notamment les systèmes de stockage NetApp ONTAP et Element, et prend également en charge les connexions NFS et iSCSI. Trident accélère le workflow DevOps en permettant aux utilisateurs d'approvisionner et de gérer le stockage à partir de leurs systèmes de stockage NetApp, sans intervention de l'administrateur de stockage.

Un administrateur peut configurer plusieurs systèmes de stockage back-end en fonction des besoins des projets et des modèles de système de stockage. Ces fonctionnalités permettent notamment la compression, des types de disques spécifiques ou des niveaux de QoS garantissant un certain niveau de performance. Une fois définis, ces systèmes back-end peuvent être utilisés par les développeurs dans leurs projets pour créer des demandes de volume persistant et connecter le stockage persistant à la demande dans leurs conteneurs.



Astra Trident a un cycle de développement rapide, et comme Kubernetes, est lancé quatre fois par an.

La dernière version d'Astra Trident est 22.01 publiée en janvier 2022. Une matrice de prise en charge pour quelle version de Trident a été testée avec laquelle une distribution Kubernetes est disponible "[ici](#)".

Depuis la version 20.04, l'opérateur Trident effectue la configuration de Trident. L'opérateur facilite les déploiements à grande échelle et offre un support supplémentaire, notamment l'auto-rétablissement des pods déployés dans le cadre de l'installation de Trident.

Avec la version 21.01, un graphique Helm a été disponible pour faciliter l'installation de l'opérateur Trident.

Téléchargez Astra Trident

Pour installer Trident sur le cluster utilisateur déployé et provisionner un volume persistant, procédez comme suit :

1. Téléchargez l'archive d'installation sur la station de travail d'administration et extrayez son contenu. La version actuelle de Trident est la version 22.01, que vous pouvez télécharger "[ici](#)".

```
[netapp-user@rhel7 ~]$ wget
https://github.com/NetApp/trident/releases/download/v22.01.0/trident-
installer-22.01.0.tar.gz
--2021-05-06 15:17:30--
https://github.com/NetApp/trident/releases/download/v22.01.0/trident-
installer-22.01.0.tar.gz
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-
releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-
98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fus-east-
1%2Ffs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-
Amz-
Signature=8a49a2a1e08c147d1ddd8149ce45a5714f9853fee19bb1c507989b9543eb36
30&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
22.01.0.tar.gz&response-content-type=application%2Foctet-stream
[following]
--2021-05-06 15:17:30-- https://github-
releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-
98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fus-east-
1%2Ffs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-
Amz-
Signature=8a49a2a1e08c147d1ddd8149ce45a5714f9853fee19bb1c507989b9543eb36
30&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
```

```

22.01.0.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-releases.githubusercontent.com (github-
releases.githubusercontent.com)... 185.199.108.154, 185.199.109.154,
185.199.110.154, ...
Connecting to github-releases.githubusercontent.com (github-
releases.githubusercontent.com)|185.199.108.154|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38349341 (37M) [application/octet-stream]
Saving to: 'trident-installer-22.01.0.tar.gz'

100%[=====
=====>] 38,349,341  88.5MB/s
in 0.4s

2021-05-06 15:17:30 (88.5 MB/s) - 'trident-installer-22.01.0.tar.gz'
saved [38349341/38349341]

```

2. Extrayez l'installation de Trident du bundle téléchargé.

```

[netapp-user@rhel7 ~]$ tar -xzf trident-installer-22.01.0.tar.gz
[netapp-user@rhel7 ~]$ cd trident-installer/
[netapp-user@rhel7 trident-installer]$

```

Installer l'opérateur Trident avec Helm

1. Définissez tout d'abord l'emplacement du cluster utilisateur `kubeconfig` Fichier en tant que variable d'environnement pour que vous n'ayez pas à le référencer, car Trident n'a pas d'option pour transmettre ce fichier.

```

[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/.ocp-
install/auth/kubeconfig

```

2. Lancer la commande Helm pour installer l'opérateur Trident à partir du tarball dans le répertoire Helm lors de la création du namespace trident dans le cluster utilisateur.

```
[netapp-user@rhel7 trident-installer]$ helm install trident
helm/trident-operator-22.01.0.tgz --create-namespace --namespace trident
NAME: trident
LAST DEPLOYED: Fri May  7 12:54:25 2021
NAMESPACE: trident
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thank you for installing trident-operator, which will deploy and manage
NetApp's Trident CSI
storage provisioner for Kubernetes.

Your release is named 'trident' and is installed into the 'trident'
namespace.
Please note that there must be only one instance of Trident (and
trident-operator) in a Kubernetes cluster.

To configure Trident to manage storage resources, you will need a copy
of tridentctl, which is
available in pre-packaged Trident releases. You may find all Trident
releases and source code
online at https://github.com/NetApp/trident.

To learn more about the release, try:

$ helm status trident
$ helm get all trident
```

3. Vous pouvez vérifier que Trident est correctement installé en vérifiant les pods qui s'exécutent dans l'espace de noms ou en utilisant le binaire tridentctl pour vérifier la version installée.

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
```

NAME	READY	STATUS	RESTARTS	AGE
trident-csi-5z451	1/2	Running	2	30s
trident-csi-696b685cf8-htdb2	6/6	Running	0	30s
trident-csi-b74p2	2/2	Running	0	30s
trident-csi-lrw4n	2/2	Running	0	30s
trident-operator-7c748d957-gr2gw	1/1	Running	0	36s

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
```

```
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 22.01.0       | 22.01.0       |
+-----+-----+
```



Dans certains cas, il est possible que les environnements client nécessitent la personnalisation du déploiement Trident. Dans ce cas, il est également possible d'installer manuellement l'opérateur Trident et de mettre à jour les manifestes inclus pour personnaliser le déploiement.

Installez manuellement l'opérateur Trident

1. Commencez par définir l'emplacement du cluster utilisateur kubeconfig Fichier en tant que variable d'environnement pour que vous n'ayez pas à le référencer, car Trident n'a pas d'option pour transmettre ce fichier.

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/.ocp-
install/auth/kubeconfig
```

2. Le `trident-installer` le répertoire contient des manifestes pour définir toutes les ressources requises. À l'aide des manifestes appropriés, créer le `TridentOrchestrator` définition de ressource personnalisée.

```
[netapp-user@rhel7 trident-installer]$ oc create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
customresourcedefinition.apiextensions.k8s.io/tridentorchestrators.tride
nt.netapp.io created
```

3. Si aucun n'existe, créez un espace de nom Trident dans le cluster à l'aide du manifeste fourni.

```
[netapp-user@rhel7 trident-installer]$ oc apply -f deploy/namespace.yaml
namespace/trident created
```

4. Créez les ressources requises pour le déploiement par un opérateur Trident, par exemple un ServiceAccount pour l'opérateur, un ClusterRole et ClusterRoleBinding à la ServiceAccount, un dédié PodSecurityPolicy, ou l'opérateur lui-même.

```
[netapp-user@rhel7 trident-installer]$ oc create -f deploy/bundle.yaml
serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created
```

5. Vous pouvez vérifier l'état de l'opérateur après son déploiement à l'aide des commandes suivantes :

```
[netapp-user@rhel7 trident-installer]$ oc get deployment -n trident
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
trident-operator    1/1     1             1           23s
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-operator-66f48895cc-lzczk  1/1     Running   0          41s
```

6. Une fois l'opérateur déployé, nous pouvons maintenant l'utiliser pour installer Trident. Cela nécessite la création d'un TridentOrchestrator.

```
[netapp-user@rhel7 trident-installer]$ oc create -f
deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created
[netapp-user@rhel7 trident-installer]$ oc describe torc trident
Name:                trident
Namespace:
Labels:               <none>
Annotations:          <none>
API Version:         trident.netapp.io/v1
Kind:                 TridentOrchestrator
Metadata:
  Creation Timestamp:  2021-05-07T17:00:28Z
  Generation:          1
  Managed Fields:
    API Version:       trident.netapp.io/v1
    Fields Type:       FieldsV1
    fieldsV1:
      f:spec:
        ..
        f:debug:
        f:namespace:
```

```

Manager:      kubect1-create
Operation:    Update
Time:         2021-05-07T17:00:28Z
API Version:  trident.netapp.io/v1
Fields Type:  FieldsV1
fieldsV1:
  f:status:
    .:
    f:currentInstallationParams:
      .:
      f:IPv6:
      f:autosupportHostname:
      f:autosupportImage:
      f:autosupportProxy:
      f:autosupportSerialNumber:
      f:debug:
      f:enableNodePrep:
      f:imagePullSecrets:
      f:imageRegistry:
      f:k8sTimeout:
      f:kubeletDir:
      f:logFormat:
      f:silenceAutosupport:
      f:tridentImage:
    f:message:
    f:namespace:
    f:status:
    f:version:
Manager:      trident-operator
Operation:    Update
Time:         2021-05-07T17:00:28Z
Resource Version:  931421
Self Link:
/apis/trident.netapp.io/v1/tridentorchestrators/trident
UID:          8a26a7a6-dde8-4d55-9b66-a7126754d81f
Spec:
  Debug:      true
  Namespace:  trident
Status:
  Current Installation Params:
    IPv6:          false
    Autosupport Hostname:
    Autosupport Image:      netapp/trident-autosupport:21.01
    Autosupport Proxy:
    Autosupport Serial Number:
    Debug:          true

```

```

Enable Node Prep:          false
Image Pull Secrets:
Image Registry:
k8sTimeout:                30
Kubelet Dir:               /var/lib/kubelet
Log Format:                 text
Silence Autosupport:       false
Trident Image:             netapp/trident:22.01.0
Message:                   Trident installed
Namespace:                 trident
Status:                    Installed
Version:                   v22.01.0
Events:
  Type      Reason      Age   From                                Message
  ----      -
Normal     Installing   80s   trident-operator.netapp.io         Installing
Trident
Normal     Installed   68s   trident-operator.netapp.io         Trident
installed

```

7. Vous pouvez vérifier que Trident est correctement installé en vérifiant les pods qui s'exécutent dans l'espace de noms ou en utilisant le binaire `tridentctl` pour vérifier la version installée.

```

[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-bb64c6cb4-lmd6h         6/6     Running   0           82s
trident-csi-gn59q                   2/2     Running   0           82s
trident-csi-m4szj                   2/2     Running   0           82s
trident-csi-sb9k9                   2/2     Running   0           82s
trident-operator-66f48895cc-lzczk    1/1     Running   0           2m39s

[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 22.01.0       | 22.01.0       |
+-----+-----+

```

Préparez les nœuds workers pour le stockage

NFS

La plupart des distributions Kubernetes sont fournies avec des packages et des utilitaires permettant de monter les systèmes back-end NFS installés par défaut, y compris Red Hat OpenShift.

Cependant, pour NFSv3, il n'existe aucun mécanisme pour négocier la simultanéité entre le client et le serveur. Par conséquent, le nombre maximal d'entrées de la table d'emplacements sunrpc côté client doit être synchronisé manuellement avec la valeur prise en charge sur le serveur pour assurer les meilleures performances de la connexion NFS sans que le serveur n'ait à diminuer la taille de la fenêtre de la connexion.

Pour ONTAP, le nombre maximal d'entrées de la table des emplacements sunrpc pris en charge est de 128, c'est-à-dire que ONTAP peut traiter 128 requêtes NFS simultanées à la fois. Cependant, par défaut, Red Hat CoreOS/Red Hat Enterprise Linux possède au maximum 65,536 entrées de table sunrpc par connexion. Nous devons définir cette valeur sur 128 et cela peut être fait à l'aide de l'opérateur de configuration machine (MCO) d'OpenShift.

Pour modifier le nombre maximal d'entrées de la table d'emplacements sunrpc dans les nœuds de travail OpenShift, procédez comme suit :

1. Connectez-vous à la console Web OCP et accédez à Compute > machine configurations. Cliquez sur Créer une configuration de machine. Copiez et collez le fichier YAML, puis cliquez sur Créer.

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 98-worker-nfs-rpc-slot-tables
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
        - contents:
            source: data:text/plain;charset=utf-8;base64,b3B0aW9ucyBzdW5ycGMgdGNwX2lheF9zbG90X3RhYmxlX2VudHJpZXM9MTI4Cg==
            filesystem: root
            mode: 420
            path: /etc/modprobe.d/sunrpc.conf
```

2. Une fois le MCO créé, la configuration doit être appliquée à tous les nœuds workers et redémarrée un par un. Le processus prend entre 20 et 30 minutes environ. Vérifiez si la configuration de la machine est appliquée à l'aide de `oc get mcp` et assurez-vous que le pool de configuration de la machine pour les employés est mis à jour.


```
[netapp-user@rhel7 openshift-deploy]$ oc get mcp
```

NAME	CONFIG	UPDATED	UPDATING
DEGRADED			
master	rendered-master-a520ae930e1d135e0dee7168	True	False
False			
worker	rendered-worker-de321b36eeba62df41feb7bc	True	False
False			

ISCSI

Pour préparer les nœuds workers afin de permettre le mappage des volumes de stockage en mode bloc via le protocole iSCSI, vous devez installer les packages nécessaires pour prendre en charge cette fonctionnalité.

Dans Red Hat OpenShift, ces opérations sont gérées via l'application d'un MCO (opérateur de configuration de machine) à votre cluster après son déploiement.

Pour configurer les nœuds workers pour exécuter des services iSCSI, procédez comme suit :

1. Connectez-vous à la console Web OCP et accédez à Compute > machine configurations. Cliquez sur Créer une configuration de machine. Copiez et collez le fichier YAML, puis cliquez sur Créer.

Lorsque vous n'utilisez pas les chemins d'accès multiples :

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-worker-element-iscsi
spec:
  config:
    ignition:
      version: 3.2.0
    systemd:
      units:
        - name: iscsid.service
          enabled: true
          state: started
  osImageURL: ""
```

Lorsque vous utilisez les chemins d'accès multiples :

```

apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 99-worker-ontap-iscsi
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
      - contents:
          source: data:text/plain;charset=utf-8;base64,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXMgbm8KICAgICAgICBmaW5kX211bHRpcGF0aHMgbm8KfQoKYmxhY2tsaXN0X2V4Y2VwdGlvbnMgewogICAgICAgIHByb3BlcnR5ICIoU0NTSV9JREVOVF98SURfV1dOKSfQoKYmxhY2tsaXN0IHsKfQoK
          verification: {}
        filesystem: root
        mode: 400
        path: /etc/multipath.conf
    systemd:
      units:
      - name: iscsid.service
        enabled: true
        state: started
      - name: multipathd.service
        enabled: true
        state: started
  osImageURL: ""

```

2. Une fois la configuration créée, il faut environ 20 à 30 minutes pour appliquer la configuration aux nœuds worker et les recharger. Vérifiez si la configuration de la machine est appliquée à l'aide de `oc get mcp` et assurez-vous que le pool de configuration de la machine pour les employés est mis à jour. Vous pouvez également vous connecter aux nœuds workers pour vérifier que le service `iscsid` est en cours d'exécution (et que le service `multipathd` est exécuté en cas d'utilisation de chemins d'accès multiples).

```
[netapp-user@rhel7 openshift-deploy]$ oc get mcp
NAME          CONFIG                                UPDATED    UPDATING
DEGRADED
master    rendered-master-a520ae930e1d135e0dee7168    True      False
False
worker    rendered-worker-de321b36eeba62df41feb7bc    True      False
False

[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled;
   vendor preset: disabled)
   Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
     Docs: man:iscsid(8)
           man:iscsiadm(8)
  Main PID: 1242 (iscsid)
    Status: "Ready to process requests"
     Tasks: 1
   Memory: 4.9M
      CPU: 9ms
   CGroup: /system.slice/iscsid.service
           └─1242 /usr/sbin/iscsid -f

[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status multipathd
● multipathd.service - Device-Mapper Multipath Device Controller
   Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled;
   vendor preset: enabled)
   Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
  Main PID: 918 (multipathd)
    Status: "up"
     Tasks: 7
   Memory: 13.7M
      CPU: 57ms
   CGroup: /system.slice/multipathd.service
           └─918 /sbin/multipathd -d -s
```



Il est également possible de confirmer que la MachineConfig a été appliquée avec succès et que les services ont été lancés comme prévu en exécutant le `oc debug` commande avec les indicateurs appropriés.

Création de systèmes back-end de stockage

Une fois l'installation d'Astra Trident Operator, vous devez configurer le système back-end pour la plateforme de stockage NetApp spécifique que vous utilisez. Suivre les liens ci-dessous pour poursuivre l'installation et la configuration d'Astra Trident.

- ["NetApp ONTAP NFS"](#)
- ["ISCSI NetApp ONTAP"](#)
- ["ISCSI NetApp Element"](#)

Configuration NetApp ONTAP NFS

Pour activer l'intégration de Trident avec le système de stockage NetApp ONTAP, il faut créer un back-end permettant la communication avec le système de stockage.

1. Des exemples de fichiers backend sont disponibles dans l'archive d'installation téléchargée dans le `sample-input` hiérarchie des dossiers. Pour les systèmes NetApp ONTAP servant de protocole NFS, copiez le `backend-ontap-nas.json` dans votre répertoire de travail et modifiez le fichier.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-samples/ontap-nas/backend-ontap-nas.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-nas.json
```

2. Modifier le `backendName`, la gestion LIF, `dataLIF`, `svm`, nom d'utilisateur, et les valeurs de mot de passe dans ce fichier.

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nas+10.61.181.221",
  "managementLIF": "172.21.224.201",
  "dataLIF": "10.61.181.221",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "password"
}
```



Il est recommandé de définir la valeur `backendName` personnalisée comme combinaison du `storageDriverName` et de la `dataLIF` qui sert NFS pour une identification facile.

3. Lorsque ce fichier backend est en place, exécutez la commande suivante pour créer votre premier back-end.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-nas.json
```

NAME	STATE	VOLUMES	STORAGE DRIVER	UUID
ontap-nas+10.61.181.221	online	0	ontap-nas	be7a619d-c81d-445c-b80c-5c87a73c5b1e

4. Lorsque le back-end est créé, vous devez ensuite créer une classe de stockage. Tout comme pour le back-end, il existe un exemple de fichier de classe de stockage qui peut être modifié pour l'environnement disponible dans le dossier des échantillons-entrées. Copiez-le dans le répertoire de travail et apportez les modifications nécessaires pour refléter le back-end créé.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. La seule modification à effectuer dans ce fichier consiste à définir le `backendType` valeur du nom du pilote de stockage du back-end nouvellement créé. Notez également la valeur `nom-champ`, qui doit être référencée ultérieurement.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
```



Il y a un champ facultatif appelé `fsType` qui est défini dans ce fichier. Cette ligne peut être supprimée dans les systèmes back-end NFS.

6. Exécutez le `oc` pour créer la classe de stockage.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

- Une fois la classe de stockage créée, vous devez ensuite créer la première demande de volume persistant. Il y a un échantillon `pvc-basic.yaml` fichier qui peut être utilisé pour effectuer cette action également située dans les entrées d'échantillons.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

- La seule modification à effectuer dans ce fichier est de s'assurer que `storageClassName` le champ correspond à celui que vous venez de créer. La définition du volume persistant peut être personnalisée davantage selon les besoins de la charge de travail à provisionner.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

- Créez le PVC en émettant le `oc` commande. La création peut prendre un certain temps en fonction de la taille du volume de sauvegarde en cours de création, de sorte que vous pouvez regarder le processus au fur et à mesure qu'il se termine.

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME      STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
basic      Bound       pvc-b4370d37-0fa4-4c17-bd86-94f96c94b42d  1Gi
RWO                                     basic-csi      7s
```

Configuration ONTAP iSCSI de NetApp

Pour activer l'intégration de Trident avec le système de stockage NetApp ONTAP, il faut créer un back-end permettant la communication avec le système de stockage.

- Des exemples de fichiers backend sont disponibles dans l'archive d'installation téléchargée dans le `sample-input` hiérarchie des dossiers. Pour les systèmes NetApp ONTAP servant iSCSI, copiez le `backend-ontap-san.json` dans votre répertoire de travail et modifiez le fichier.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/ontap-san/backend-ontap-san.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-san.json
```

2. Modifiez les valeurs LIF, dataLIF, svm, nom d'utilisateur et mot de passe dans ce fichier.

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "172.21.224.201",
  "dataLIF": "10.61.181.240",
  "svm": "trident_svm",
  "username": "admin",
  "password": "password"
}
```

3. Lorsque ce fichier backend est en place, exécutez la commande suivante pour créer votre premier backend.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-san.json
+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE | VOLUMES |          |          |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontapsan_10.61.181.241 | ontap-san      | 6788533c-7fea-4a35-b797- |
| fb9bb3322b91 | online |          0 |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

4. Lorsque le back-end est créé, vous devez ensuite créer une classe de stockage. Tout comme pour le back-end, il existe un exemple de fichier de classe de stockage qui peut être modifié pour l'environnement disponible dans le dossier des échantillons-entrées. Copiez-le dans le répertoire de travail et apportez les modifications nécessaires pour refléter le back-end créé.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. La seule modification à effectuer dans ce fichier consiste à définir le backendType valeur du nom du pilote de stockage du back-end nouvellement créé. Notez également la valeur nom-champ, qui doit être

référéncée ultérieurement.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
```



Il y a un champ facultatif appelé `fsType` qui est défini dans ce fichier. Dans les systèmes back-end iSCSI, cette valeur peut être définie sur un type de système de fichiers Linux spécifique (XFS, ext4, etc.) ou peut être supprimée pour permettre à OpenShift de décider du système de fichiers à utiliser.

6. Exécutez le `oc` pour créer la classe de stockage.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

7. Une fois la classe de stockage créée, vous devez ensuite créer la première demande de volume persistant. Il y a un échantillon `pvc-basic.yaml` fichier qui peut être utilisé pour effectuer cette action également située dans les entrées d'échantillons.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. La seule modification à effectuer dans ce fichier est de s'assurer que `storageClassName` le champ correspond à celui que vous venez de créer. La définition du volume persistant peut être personnalisée davantage selon les besoins de la charge de travail à provisionner.


```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

9. Créez le PVC en émettant le `oc` commande. La création peut prendre un certain temps en fonction de la taille du volume de sauvegarde en cours de création, de sorte que vous pouvez regarder le processus au fur et à mesure qu'il se termine.

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
```

NAME	STATUS	VOLUME	CAPACITY
ACCESS MODES	STORAGECLASS	AGE	
basic	Bound	pvc-7ceac1ba-0189-43c7-8f98-094719f7956c	1Gi
RWO		basic-csi	3s

Configuration iSCSI de NetApp Element

Pour activer l'intégration de Trident avec le système de stockage NetApp Element, vous devez créer un backend permettant la communication avec le système de stockage via le protocole iSCSI.

1. Des exemples de fichiers backend sont disponibles dans l'archive d'installation téléchargée dans le `sample-input` hiérarchie des dossiers. Pour les systèmes NetApp Element servant iSCSI, copiez le `backend-solidfire.json` dans votre répertoire de travail et modifiez le fichier.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/solidfire/backend-solidfire.json ./
[netapp-user@rhel7 trident-installer]$ vi ./backend-solidfire.json
```

- a. Modifiez l'utilisateur, le mot de passe et la valeur MVIP sur le `EndPoint` ligne.
- b. Modifiez le `SVIP` valeur.

```
{
  "version": 1,
  "storageDriverName": "solidfire-san",
  "Endpoint": "https://trident:password@172.21.224.150/json-
rpc/8.0",
  "SVIP": "10.61.180.200:3260",
  "TenantName": "trident",
  "Types": [{"Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS":
2000, "burstIOPS": 4000}},
            {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS":
6000, "burstIOPS": 8000}},
            {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS":
8000, "burstIOPS": 10000}}]
}
```

2. Avec ce fichier back-end en place, exécutez la commande suivante pour créer votre premier back-end.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-solidfire.json
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE | VOLUMES | |          |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| solidfire_10.61.180.200 | solidfire-san  | b90783ee-e0c9-49af-8d26-
3ea87ce2efdf | online |          0 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

3. Lorsque le back-end est créé, vous devez ensuite créer une classe de stockage. Tout comme pour le back-end, il existe un exemple de fichier de classe de stockage qui peut être modifié pour l'environnement disponible dans le dossier des échantillons-entrées. Copiez-le dans le répertoire de travail et apportez les modifications nécessaires pour refléter le back-end créé.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.tmpl ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

4. La seule modification à effectuer dans ce fichier consiste à définir le backendType valeur du nom du pilote de stockage du back-end nouvellement créé. Notez également la valeur nom-champ, qui doit être référencée ultérieurement.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "solidfire-san"

```



Il y a un champ facultatif appelé `fsType` qui est défini dans ce fichier. Dans les systèmes back-end iSCSI, cette valeur peut être définie sur un type de système de fichiers Linux spécifique (XFS, ext4, etc.), ou elle peut être supprimée pour permettre à OpenShift de décider du système de fichiers à utiliser.

5. Exécutez le `oc` pour créer la classe de stockage.

```

[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-basic.yaml
storageclass.storage.k8s.io/basic-csi created

```

6. Une fois la classe de stockage créée, vous devez ensuite créer la première demande de volume persistant. Il y a un échantillon `pvc-basic.yaml` fichier qui peut être utilisé pour effectuer cette action également située dans les entrées d'échantillons.

```

[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml

```

7. La seule modification à effectuer dans ce fichier est de s'assurer que `storageClassName` le champ correspond à celui que vous venez de créer. La définition du volume persistant peut être personnalisée davantage selon les besoins de la charge de travail à provisionner.

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi

```

8. Créez le PVC en émettant le `oc` commande. La création peut prendre un certain temps en fonction de la taille du volume de sauvegarde en cours de création, de sorte que vous pouvez regarder le processus au fur et à mesure qu'il se termine.

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
```

NAME	STATUS	VOLUME	CAPACITY
basic	Bound	pvc-3445b5cc-df24-453d-a1e6-b484e874349d	1Gi
		basic-csi	5s

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 DÉLICTELLE (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.