Cloud hybride avec composants gérés par le fournisseur

NetApp Solutions

NetApp
April 23, 2024

This PDF was generated from https://docs.netapp.com/fr-fr/netapp-solutions/rhhc/rhhc-pm-solution.html on April 23, 2024. Always check docs.netapp.com for the latest.

Sommaire

Solutions multicloud hybrides NetApp pour les workloads de conteneurs Red Hat OpenShift 1
Présentation 1
Solution NetApp avec les workloads gérés de la plateforme de conteneurs Red Hat OpenShift sur AWS . . . 3
Déployez et configurez la plateforme de conteneurs Red Hat OpenShift gérée sur AWS 4
Protection des données 7
Migration des données 23

Solutions multicloud hybrides NetApp pour les workloads de conteneurs Red Hat OpenShift

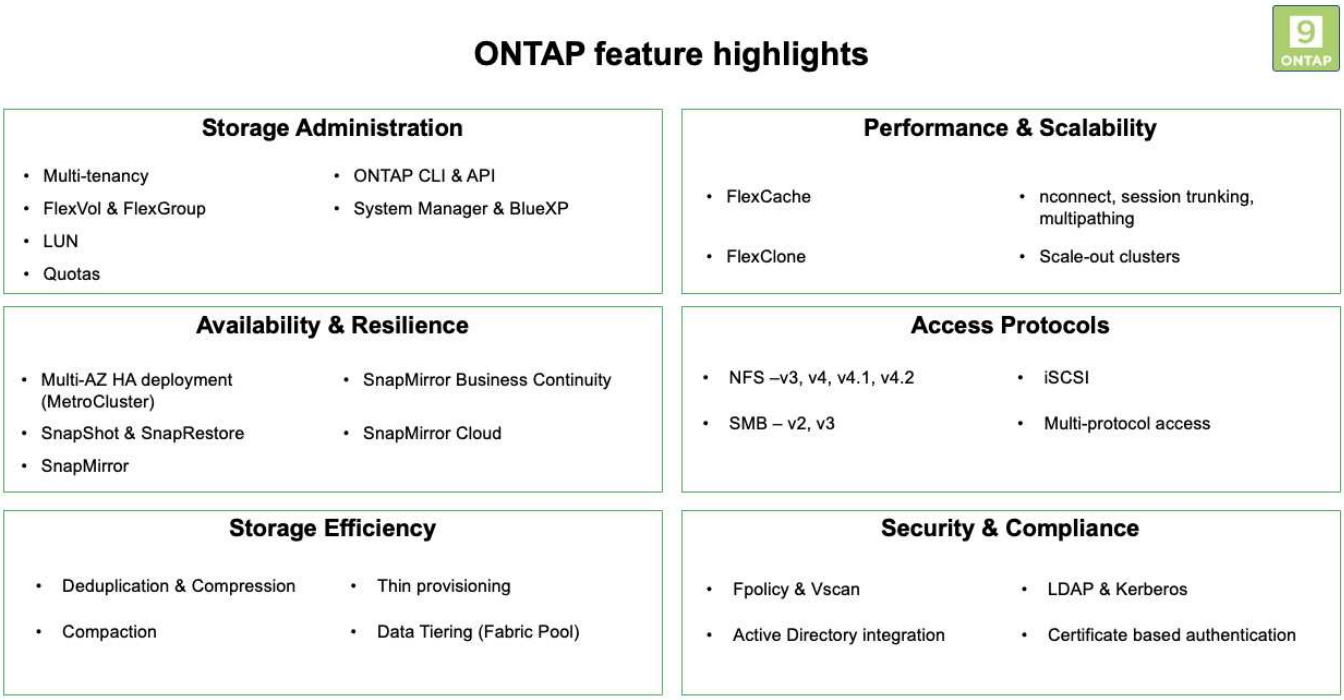
Présentation

NetApp constate une augmentation significative des clients qui modernisent leurs applications d'entreprise existantes et créent de nouvelles applications à l'aide de conteneurs et de plateformes d'orchestration basées sur Kubernetes. Nous avons adopté Red Hat OpenShift Container Platform comme bon nombre de nos clients.

Alors que les clients sont de plus en plus nombreux à adopter des conteneurs dans leur entreprise, NetApp est parfaitement positionné pour répondre aux besoins de stockage persistant de leurs applications avec état et aux besoins de gestion des données classiques, tels que la protection, la sécurité et la migration des données. Toutefois, ces besoins sont satisfaits à l'aide de stratégies, d'outils et de méthodes différents.

Les options de stockage basées sur NetApp ONTAP sont énumérées ci-dessous et offrent sécurité, protection des données, fiabilité et flexibilité pour les conteneurs et les déploiements Kubernetes.

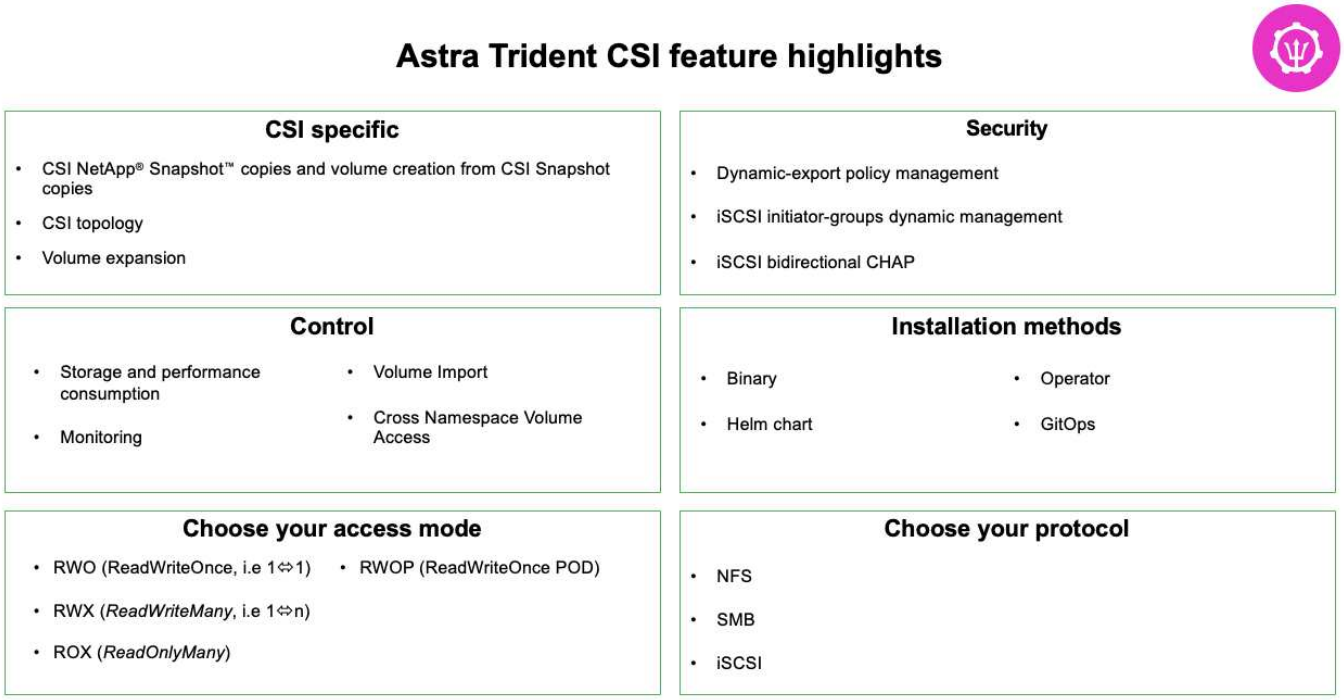
- Stockage autogéré sur site :
  - Stockage FAS (Fabric Attached Storage), baies FAS 100 % Flash (AFF), baies SAN ASA (All SAN Array) et ONTAP Select
- Stockage géré par un fournisseur sur site :
  - NetApp Keystone fournit une solution de stockage en tant que service (STaaS)
- Stockage autogéré dans le cloud :
  - NetApp Cloud volumes ONTAP (CVO) fournit un stockage autogéré dans les hyperscalers
- Stockage géré par un fournisseur dans le cloud :
  - Cloud Volumes Service pour Google Cloud (CVS), Azure NetApp Files (ANF) et Amazon FSX pour NetApp ONTAP offrent un stockage entièrement géré dans les hyperscalers



NetApp BlueXP vous permet de gérer l'ensemble de vos ressources de stockage et de données à partir d'un seul plan de contrôle/interface.

Vous pouvez utiliser BlueXP pour créer et gérer du stockage cloud (par exemple, Cloud Volumes ONTAP et Azure NetApp Files), déplacer, protéger et analyser les données, et contrôler de nombreux systèmes de stockage sur site et en périphérie.

NetApp Astra Trident est un orchestrateur de stockage conforme à CSI qui permet de consommer rapidement et facilement du stockage persistant grâce à plusieurs options de stockage NetApp mentionnées ci-dessus. Il s'agit d'un logiciel open source géré et pris en charge par NetApp.



Les workloads de conteneurs stratégiques requièrent bien plus que de simples volumes persistants. Leurs exigences de gestion des données requièrent également la protection et la migration des objets kubernetes applicatifs.

> Les données d'application incluent des objets kubernetes en plus des données utilisateur. Voici quelques exemples : - Objets kubernetes tels que les pods Specs, les PVC, les déploiements, les services - objets de configuration personnalisés tels que les cartes de configuration et les secrets - données persistantes telles que les copies Snapshot, les sauvegardes, les clones - ressources personnalisées telles que CRS et CRD

NetApp Astra Control, disponible en tant que logiciel entièrement géré et autogéré, assure l'orchestration pour une gestion robuste des données d'application. Reportez-vous à la "Documentation Astra" Pour en savoir plus sur la gamme de produits Astra.

Cette documentation de référence apporte la validation de la migration et de la protection des applications basées sur des conteneurs, déployées sur une plateforme de conteneurs RedHat OpenShift à l'aide de NetApp Astra Control Center. En outre, la solution fournit des détails généraux sur le déploiement et l'utilisation de Red Hat Advanced Cluster Management (ACM) pour la gestion des plateformes de conteneurs. Ce document détaille également les modalités d'intégration du stockage NetApp avec les plateformes de conteneurs Red Hat OpenShift à l'aide d'Astra Trident CSI Provisioner. ASTRA Control Center est déployé sur le cluster Hub et est utilisé pour gérer les applications de conteneur et leur cycle de vie de stockage persistant.

Enfin, il fournit une solution de réplication, de basculement et de retour arrière pour les workloads de conteneurs sur des clusters Red Hat OpenShift gérés dans AWS (ROSA) utilisant Amazon FSX pour NetApp ONTAP (FSxN) en tant que stockage persistant.

Solution NetApp avec les workloads gérés de la plateforme de conteneurs Red Hat OpenShift sur AWS

Solution NetApp avec les workloads gérés de la plateforme de conteneurs Red Hat OpenShift sur AWS

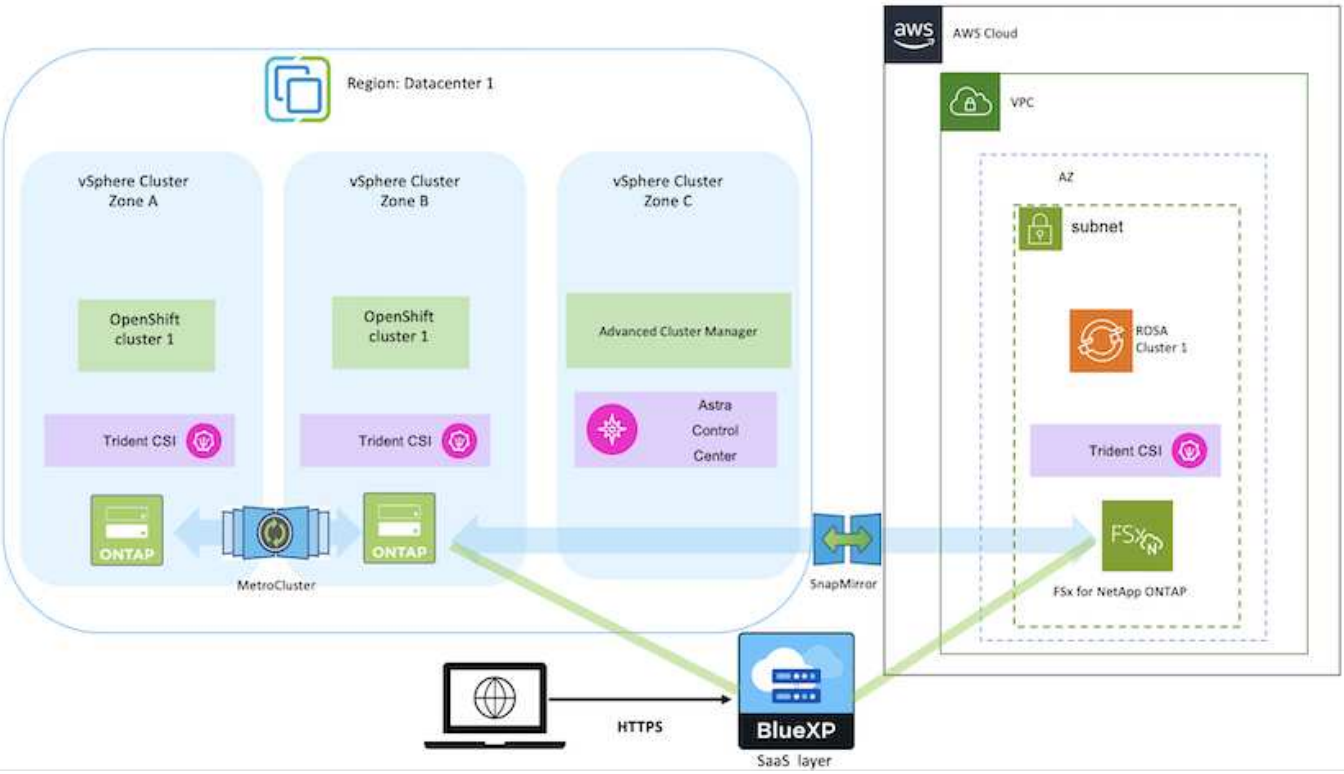
Les clients peuvent être « nés dans le cloud » ou se trouver à un point de leur parcours de modernisation lorsqu'ils sont prêts à déplacer des workloads spécifiques ou tous les workloads de leurs data centers vers le cloud. Ils peuvent choisir d'utiliser des conteneurs OpenShift gérés par un fournisseur et du stockage NetApp géré par un fournisseur dans le cloud pour exécuter leurs workloads. Ils doivent planifier et déployer les clusters de conteneurs Red Hat OpenShift (ROSA) gérés dans le cloud pour assurer la réussite de leur environnement de production pour leurs workloads de conteneurs. Dans le cloud AWS, ils peuvent également déployer FSX pour NetApp ONTAP pour répondre aux besoins en stockage.

FSX pour NetApp ONTAP assure la protection, la fiabilité et la flexibilité des données pour les déploiements de conteneurs dans AWS. ASTRA Trident sert de mécanisme de provisionnement de stockage dynamique pour consommer le stockage FSxN persistant pour les applications avec état des clients.

COMME ROSA peut être déployée en mode HA avec des nœuds de plan de contrôle répartis sur plusieurs zones de disponibilité, FSX ONTAP peut également être provisionné avec l'option multi-AZ qui assure la haute disponibilité et la protection contre les défaillances AZ.

> L'accès à un système de fichiers Amazon FSX à partir de la zone de disponibilité préférée (AZ) du système de fichiers ne comporte aucun frais de transfert de données. Pour plus d'informations sur les prix, reportez-vous à la section "ici".

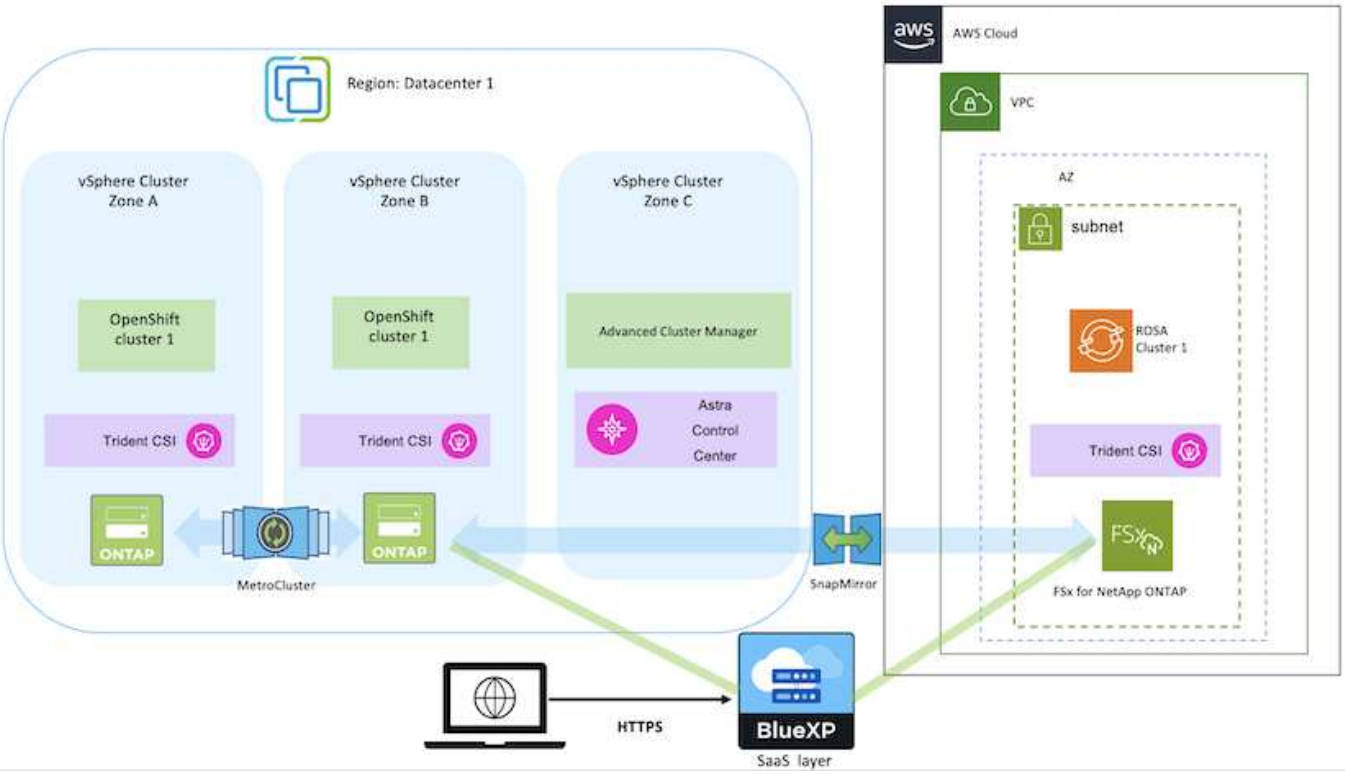
Solution de protection et de migration des données pour les workloads de conteneurs OpenShift



Déployez et configurez la plateforme de conteneurs Red Hat OpenShift gérée sur AWS

Cette section décrit un workflow général de configuration des clusters Red Hat OpenShift gérés sur AWS(ROSA). Il présente l'utilisation de FSX for NetApp ONTAP (FSxN) géré en tant que back-end de stockage par Astra Trident pour fournir des volumes persistants. Vous trouverez des informations détaillées sur le déploiement de FSxN sur AWS à l'aide de BlueXP. Vous y trouverez également des informations détaillées sur l'utilisation de BlueXP et d'OpenShift GitOps (Argo CD) pour exécuter les activités de protection et de migration des données pour les applications avec état sur les clusters ROSA.

Voici un diagramme illustrant les clusters ROSA déployés sur AWS et utilisant FSxN comme stockage back-end.



Cette solution a été vérifiée en utilisant deux clusters ROSA dans deux VPC dans AWS. Chaque cluster ROSA a été intégré à FSxN à l'aide d'Astra Trident. IL existe plusieurs façons de déployer les clusters ROSA et FSxN dans AWS. Cette description de haut niveau de la configuration fournit des liens de documentation pour la méthode spécifique qui a été utilisée. Vous pouvez vous référer aux autres méthodes dans les liens pertinents fournis dans le "ressources".

Le processus de configuration peut être divisé en plusieurs étapes :

Installer les clusters ROSA

- Créez deux VPC et configurez la connectivité de peering VPC entre les VPC.
- Reportez-vous à "ici" Pour obtenir des instructions sur l'installation des clusters ROSA.

Installez FSxN

- Installez FSxN sur les VPC de BlueXP. Reportez-vous à "ici" Pour créer un compte BlueXP et démarrer. Reportez-vous à "ici" Pour l'installation de FSxN. Reportez-vous à "ici" Pour créer un connecteur dans AWS pour gérer le FSxN.
- Déploiement de FSxN à l'aide d'AWS Reportez-vous à "ici" Déploiement via la console AWS

Installation de Trident sur les clusters ROSA (à l'aide du graphique Helm)

- Utilisez le tableau Helm pour installer Trident sur les clusters ROSA. url du graphique Helm : https://netapp.github.io/trident-helm-chart

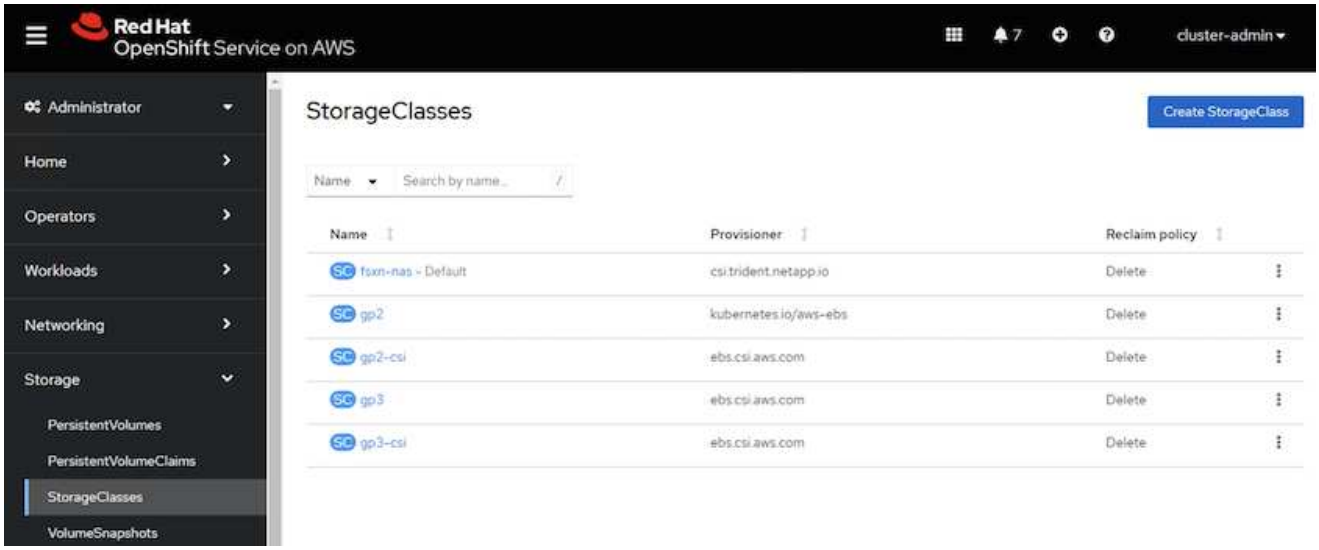
Intégration de FSxN avec Astra Trident pour les clusters ROSA

OpenShift GitOps peut être utilisé pour déployer Astra Trident CSI sur tous les clusters gérés lors de leur enregistrement sur ArgoCD à l'aide d'ApplicationSet.

```yaml
apiVersion: argoproj.io/v1alpha1
kind: ApplicationSet
metadata:
  name: trident-operator
spec:
  generators:
  - clusters: {}
    # selector:
    #   matchLabels:
    #     tridentversion: '23.04.0'
  template:
    metadata:
      name: '{{nameNormalized}}-trident'
    spec:
      destination:
        namespace: trident
        server: '{{server}}'
      source:
        repoURL: 'https://netapp.github.io/trident-helm-chart'
        targetRevision: 23.04.0
        chart: trident-operator
      project: default
      syncPolicy:
        syncOptions:
          - CreateNamespace=true
```

Création de classes de stockage et de back-end à l'aide de Trident (pour FSxN)

- Reportez-vous à "ici" pour plus d'informations sur la création de systèmes back-end et de classes de stockage.
- Créez la classe de stockage créée pour FsxN avec Trident CSI par défaut depuis la console OpenShift. Voir la capture d'écran ci-dessous :



Déployer une application à l'aide d'OpenShift GitOps (Argo CD)

- Installez l'opérateur OpenShift GitOps sur le cluster. Reportez-vous aux instructions "ici".
- Configurez une nouvelle instance Argo CD pour le cluster. Reportez-vous aux instructions "ici".

Ouvrez la console du CD Argo et déployez une application. Par exemple, vous pouvez déployer une application Jenkins à l'aide du CD Argo avec Helm Chart. Lors de la création de l'application, les détails suivants ont été fournis : projet : cluster par défaut : https://kubernetes.default.svcEspace de noms : Jenkins l'url du graphique Helm : https://charts.bitnami.com/bitnami

Paramètres Helm : global.storageClass : fsxn-nas

Protection des données

Cette page présente les options de protection des données pour les clusters Red Hat OpenShift sur AWS (ROSA) gérés à l'aide d'Astra Control Service. ASTRA Control Service (ACS) est une interface utilisateur graphique simple d'utilisation qui vous permet d'ajouter des clusters, de définir des applications qui s'exécutent sur eux et d'effectuer des activités de gestion des données intégrant la cohérence applicative. Les fonctions ACS sont également accessibles via une API qui permet l'automatisation des flux de travail.

NetApp Astra Trident est le moteur d'Astra Control (ACS ou ACC). ASTRA Trident intègre plusieurs types de clusters Kubernetes tels que Red Hat OpenShift, EKS, AKS, SUSE Rancher, Anthos, etc. avec plusieurs versions de stockage NetApp ONTAP telles que FAS/AFF, ONTAP Select, CVO, Google Cloud Volumes

Service, Azure NetApp Files et Amazon FSX pour NetApp ONTAP.

Cette section fournit des détails sur les options de protection des données suivantes à l'aide d'ACS :

- Vidéo montrant la sauvegarde et la restauration d'une application ROSA s'exécutant dans une région et la restauration dans une autre région.
- Vidéo présentant Snapshot et la restauration d'une application ROSA.
- Détails détaillés de l'installation d'un cluster ROSA, Amazon FSX pour NetApp ONTAP, à l'aide d'NetApp Astra Trident pour l'intégration au back-end de stockage, installation d'une application postgresql sur un cluster ROSA, utilisation d'ACS pour créer un snapshot de l'application et restaurer son application.
- Un blog montrant les détails pas à pas de la création et de la restauration à partir d'un snapshot pour une application mysql sur un cluster ROSA avec FSX pour ONTAP à l'aide d'ACS.

Sauvegarde/Restauration à partir de la sauvegarde

La vidéo suivante montre la sauvegarde d'une application ROSA s'exécutant dans une région et la restauration dans une autre région.

FSX NetApp ONTAP pour Red Hat OpenShift Service sur AWS

Snapshot/Restaurer à partir d'un snapshot

La vidéo suivante montre la prise d'un instantané d'une application ROSA et la restauration à partir de l'instantané après.

Snapshot/Restore pour les applications sur les clusters Red Hat OpenShift Service sur AWS (ROSA) avec le stockage Amazon FSX pour NetApp ONTAP

Blog

- "Utilisation d'Astra Control Service pour la gestion des données des applications sur des clusters ROSA avec le stockage Amazon FSX"

Détails détaillés étape par étape pour créer un snapshot et le restaurer à partir de celui-ci

Configuration des prérequis

- "Compte AWS"
- "Compte Red Hat OpenShift"
- Utilisateur IAM avec "autorisations appropriées" Pour créer et accéder au cluster ROSA
- "CLI AWS"
- "CLI ROSA"
- "Interface de ligne de commande OpenShift"(oc)
- VPC avec sous-réseaux et passerelles et routes appropriées
- "ROSA Cluster installée" Dans le VPC
- "Amazon FSX pour NetApp ONTAP" Créées dans le même VPC
- Accès au cluster ROSA depuis "OpenShift Hybrid Cloud Console"

Étapes suivantes

1. Créer un utilisateur admin et se connecter au cluster
2. Créez un fichier kubeconfig pour le cluster.
3. Installer Astra Trident sur le cluster.
4. Créez une configuration back-end, une classe de stockage et une classe Snapshot à l'aide du mécanisme de provisionnement Trident CSI.
5. Déployer une application postgresql sur le cluster.
6. Créez une base de données et ajoutez un enregistrement.
7. Ajoutez le cluster dans ACS.
8. Définissez l'application dans ACS.
9. Créez un instantané à l'aide d'ACS.
10. Supprimez la base de données dans l'application postgresql.
11. Restauration à partir d'un snapshot à l'aide d'ACS.
12. Vérifiez que votre application a été restaurée à partir de l'instantané.

1. Créer un utilisateur admin et se connecter au cluster

Accédez au cluster ROSA en créant un utilisateur admin avec la commande suivante : (vous devez créer un utilisateur admin uniquement si vous n'en avez pas créé un au moment de l'installation)

```
rosa create admin --cluster=<cluster-name>
```

La commande fournit une sortie qui ressemble à ce qui suit. Connectez-vous au cluster à l'aide de `oc login` commande fournie dans la sortie.

```
W: It is recommended to add an identity provider to login to this cluster.
See 'rosa create idp --help' for more information.
I: Admin account has been added to cluster 'my-rosa-cluster'. It may take up
to a minute for the account to become active.
I: To login, run the following command:
oc login https://api.my-rosa-cluster.abcd.p1.openshiftapps.com:6443 \
--username cluster-admin \
--password FWGYL-2mkJI-00000-00000
```

> Vous pouvez également vous connecter au cluster à l'aide d'un jeton. Si vous avez déjà créé un administrateur au moment de la création du cluster, vous pouvez vous connecter au cluster depuis la console Red Hat OpenShift Hybrid Cloud à l'aide des informations d'identification de l'administrateur. Ensuite, en cliquant sur le coin supérieur droit où il affiche le nom de l'utilisateur connecté, vous pouvez obtenir le `oc login` commande (jeton de connexion) pour la ligne de commande.

2. Créez un fichier kubeconfig pour le cluster

Suivre les procédures "ici" Pour créer un fichier kubeconfig pour le cluster ROSA. Ce fichier kubeconfig sera utilisé plus tard lorsque vous ajoutez le cluster dans ACS.

3. Installer Astra Trident sur le cluster

Installer Astra Trident (dernière version) sur le cluster ROSA. Pour ce faire, vous pouvez suivre l'une des procédures données "ici". Pour installer Trident à l'aide de Helm à partir de la console du cluster, commencez par créer un projet appelé Trident.

Ensuite, dans la vue Développeur, créez un référentiel de graphiques Helm. Pour le champ URL, utilisez `'https://netapp.github.io/trident-helm-chart'`. Créez ensuite une version de Helm pour l'opérateur Trident.

Project: trident

Create Helm Chart Repository

Add helm chart repository.

Configure via: (•) Form view ( ) YAML view

Scope type

( ) Namespaced scoped (ProjectHelmChartRepository)

Add Helm Chart Repository in the selected namespace.

(•) Cluster scoped (HelmChartRepository)

Add Helm Chart Repository at the cluster level and in all namespaces.

Name *

trident

A unique name for the Helm Chart repository.

Display name

Astra Trident

A display name for the Helm Chart repository.

Description

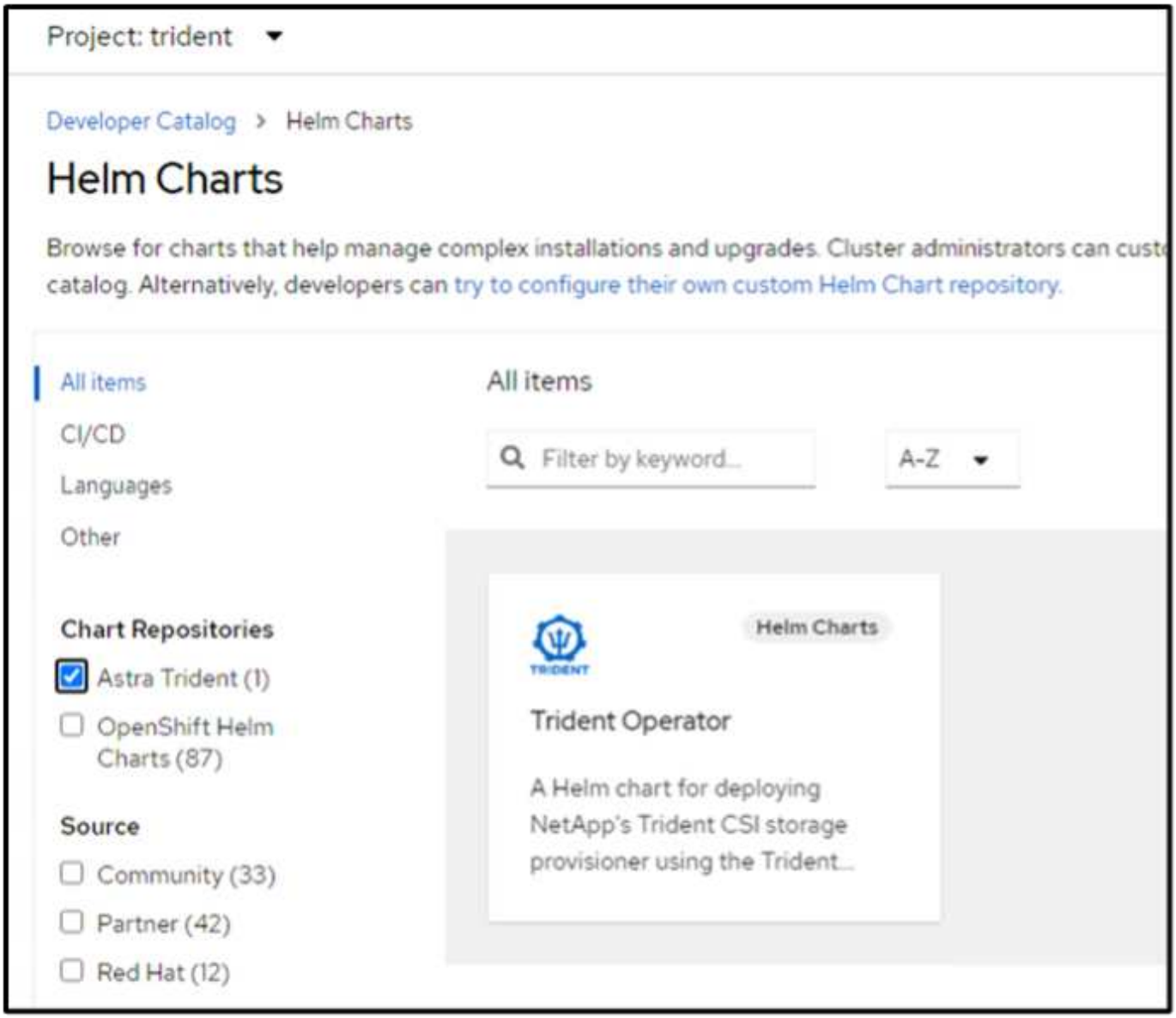
NetApp Astra Trident

A description for the Helm Chart repository.

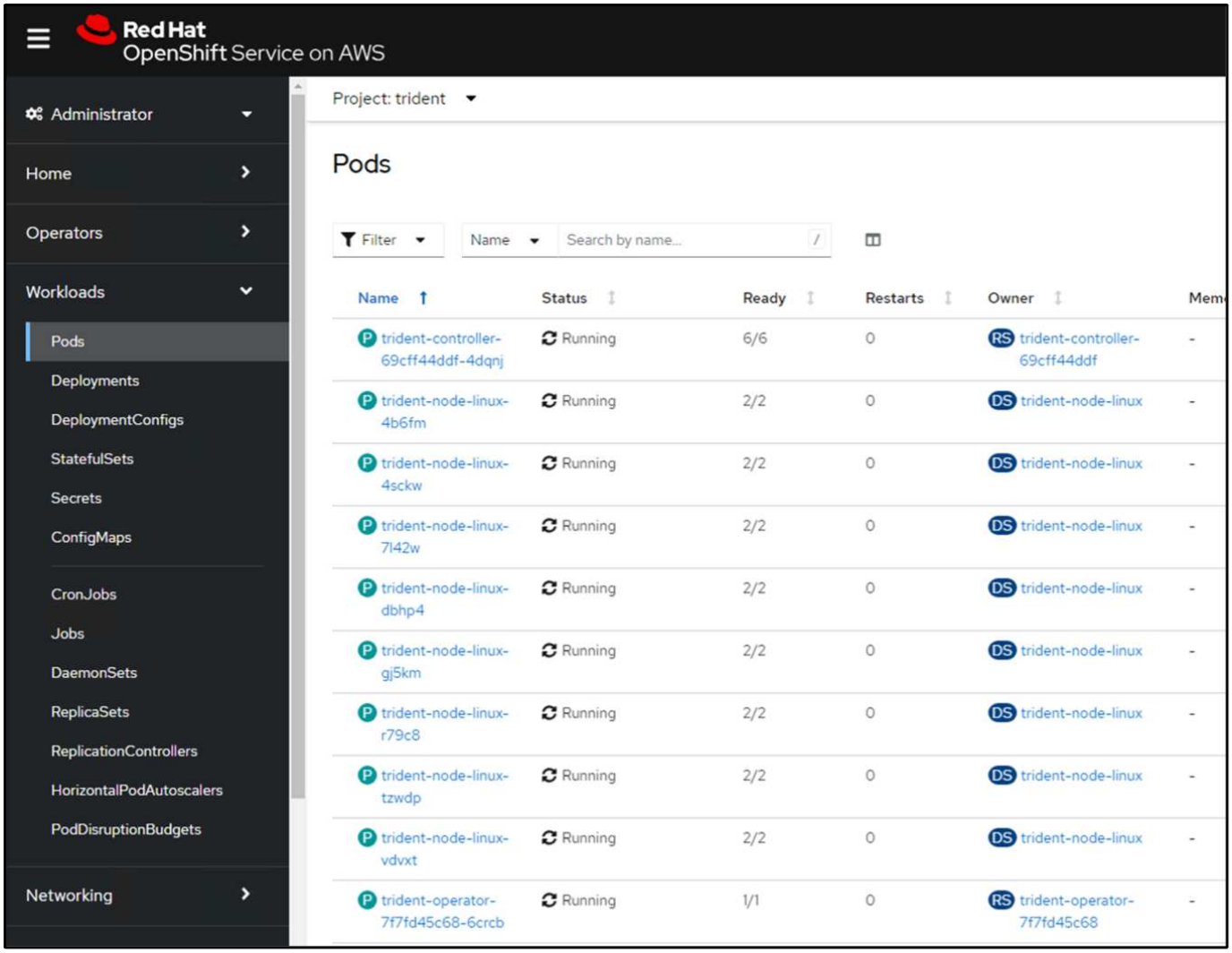
[ ] Disable usage of the repo in the developer catalog.

URL *

https://netapp.github.io/trident-helm-chart



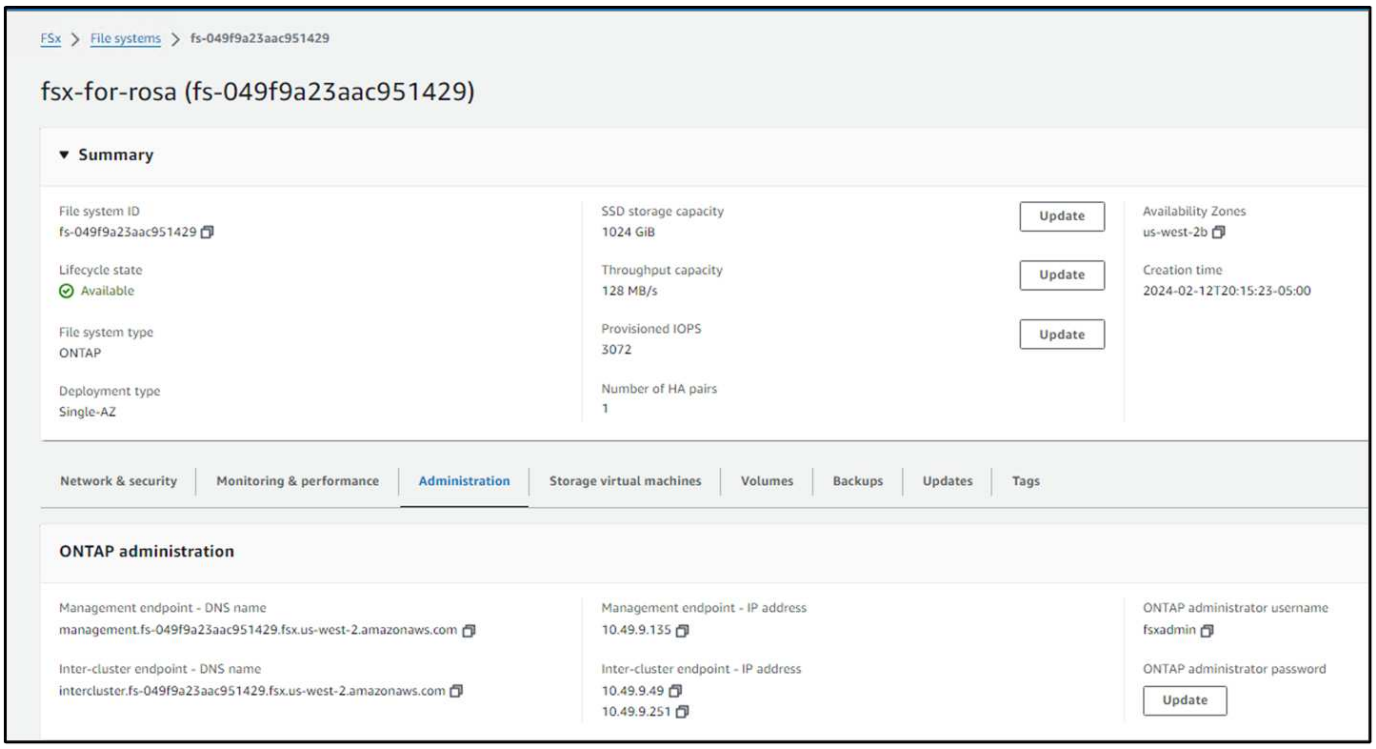
Vérifiez que tous les pods trident sont en cours d'exécution en revenant à la vue de l'administrateur de la console et en sélectionnant les pods dans le projet trident.



4. Créez une configuration backend, de classe de stockage et de classe de snapshots à l'aide du mécanisme de provisionnement Trident CSI

Utilisez les fichiers yaml illustrés ci-dessous pour créer un objet back-end trident, un objet classe de stockage et l'objet Volumesnapshot. Assurez-vous de fournir les informations d'identification de votre système de fichiers Amazon FSX for NetApp ONTAP que vous avez créé, la LIF de gestion et le nom de vServer de votre système de fichiers dans la configuration yaml pour le back-end. Pour obtenir ces informations, accédez à la console AWS pour Amazon FSX et sélectionnez le système de fichiers, accédez à l'onglet Administration. Cliquez également sur mettre à jour pour définir le mot de passe du `fsxadmin` utilisateur.

> Vous pouvez utiliser la ligne de commande pour créer les objets ou les créer avec les fichiers yaml à partir de la console de cloud hybride.



Configuration back-end Trident

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-nas-secret
type: Opaque
stringData:
  username: fsxadmin
  password: <password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: ontap-nas
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: <management lif>
  backendName: ontap-nas
  svm: fsx
  credentials:
    name: backend-tbc-ontap-nas-secret
```

Classe de stockage

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
allowVolumeExpansion: true
```

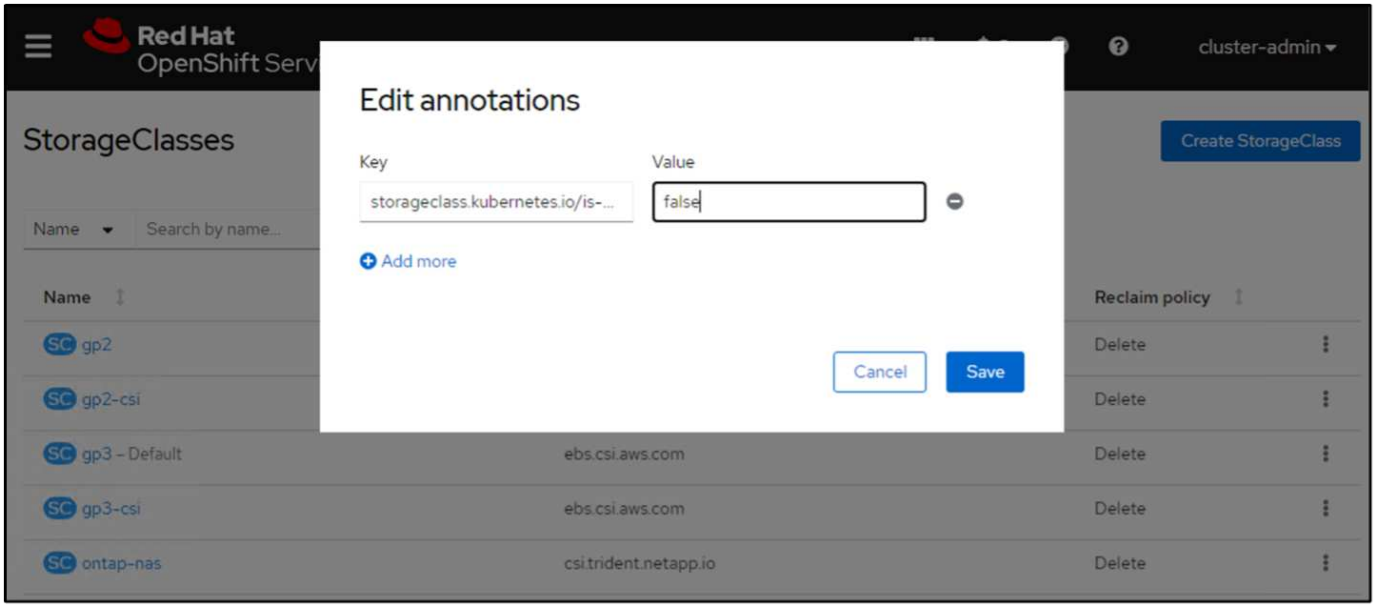
classe d'instantanés

```yaml
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: trident-snapshotclass
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

Vérifiez que le backend, la classe de stockage et les objets trident-snapshotclass sont créés à l'aide des commandes indiquées ci-dessous.

```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get tbc -n trident
NAME        BACKEND NAME   BACKEND UUID                           PHASE   STATUS
ontap-nas   ontap-nas      8a5e4583-2dac-46bb-b01e-fa7c3816f121   Bound   Success
[ec2-user@ip-10-49-11-132 storage]$ kubectl get sc
NAME            PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
gp2             kubernetes.io/aws-ebs   Delete          WaitForFirstConsumer   true                   3h23m
gp2-csi         ebs.csi.aws.com         Delete          WaitForFirstConsumer   true                   3h19m
gp3 (default)   ebs.csi.aws.com         Delete          WaitForFirstConsumer   true                   3h23m
gp3-csi         ebs.csi.aws.com         Delete          WaitForFirstConsumer   true                   3h19m
ontap-nas       csi.trident.netapp.io   Delete          Immediate              true                   141m
[ec2-user@ip-10-49-11-132 storage]$ kubectl get Volumesnapshotclass
NAME                    DRIVER                  DELETIONPOLICY   AGE
csi-aws-vsc             ebs.csi.aws.com         Delete           3h19m
trident-snapshotclass   csi.trident.netapp.io   Delete           6m56s
[ec2-user@ip-10-49-11-132 storage]$
```

Vous devez à présent modifier de façon importante : définir ontap-nas comme classe de stockage par défaut au lieu de gp3 de sorte que l'application postgresql que vous déployez ultérieurement puisse utiliser la classe de stockage par défaut. Dans la console OpenShift de votre cluster, sous stockage, sélectionnez classes de stockage. Editez l'annotation de la classe par défaut actuelle à false et ajoutez l'annotation storageclass.kubernetes.io/is-default-class set à true pour la classe de stockage ontap-nas.



5. Déployer une application postgresql sur le cluster

Vous pouvez déployer l'application à partir de la ligne de commande comme suit :

```
helm install postgresql bitnami/postgresql -n postgresql --create-namespace
```

```
[ec2-user@ip-10-49-11-132 astra]$ helm install postgresql bitnami/postgresql -n postgresql --create-namespace
NAME: postgresql
LAST DEPLOYED: Tue Feb 13 14:46:16 2024
NAMESPACE: postgresql
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: postgresql
CHART VERSION: 14.0.4
APP VERSION: 16.2.0

** Please be patient while the chart is being deployed **

PostgreSQL can be accessed via port 5432 on the following DNS names from within your cluster:

    postgresql.postgresql.svc.cluster.local - Read/Write connection

To get the password for "postgres" run:

    export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="{.data.postgres-password}" | base64 -d)

To connect to your database run the following command:

    kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image docker.io/bitnami/postgresql:16.2.0-debian-11-r1 --env="PG
PASSWORD=$POSTGRES_PASSWORD" \
      --command -- psql --host postgresql -U postgres -d postgres -p 5432

    > NOTE: If you access the container using bash, make sure that you execute "/opt/bitnami/scripts/postgresql/entrypoint.sh /bin/bash" in order to avoid
the error "psql: local user with ID 1001} does not exist"

To connect to your database from outside the cluster execute the following commands:

    kubectl port-forward --namespace postgresql svc/postgresql 5432:5432 &
    PGPASSWORD="$POSTGRES_PASSWORD" psql --host 127.0.0.1 -U postgres -d postgres -p 5432

WARNING: The configured password will be ignored on new installation in case when previous PostgreSQL release was deleted through the helm command. In that
 case, old PVC will have an old password, and setting it through helm won't take effect. Deleting persistent volumes (PVs) will solve the issue.
[ec2-user@ip-10-49-11-132 astra]$
```

Si vous ne voyez pas les modules d'application s'exécuter, une erreur peut survenir en raison de contraintes de contexte de sécurité.

```
[ec2-user@ip-10-49-11-132 astra]$ kubectl get all -n postgresql
NAME                    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/postgresql      ClusterIP   172.30.245.50   <none>        5432/TCP   12m
service/postgresql-hl   ClusterIP   None            <none>        5432/TCP   12m

NAME                          READY   AGE
statefulset.apps/postgresql   0/1     12m
[ec2-user@ip-10-49-11-132 astra]$ kubectl get events -n postgresql
LAST SEEN   TYPE      REASON                  OBJECT                                  MESSAGE
2m39s       Normal    WaitForFirstConsumer    persistentvolumeclaim/data-postgresql-0   waiting for first consumer to be created before binding
12m         Normal    SuccessfulCreate        statefulset/postgresql                  create Claim data-postgresql-0 Pod postgresql-0 in StatefulSet postg
resql success
107s        Warning   FailedCreate            statefulset/postgresql                  create Pod postgresql-0 in StatefulSet postgresql failed error: pods
 "postgresql-0" is forbidden: unable to validate against any security context constraint: [provider "trident-controller": Forbidden: not usable by user or
serviceaccount, provider "anyuid": Forbidden: not usable by user or serviceaccount, provider restricted-v2: .spec.securityContext.fsGroup: Invalid value: [
]int64{1001}: 1001 is not an allowed group, provider restricted-v2: .containers[0].runAsUser: Invalid value: 1001: must be in the ranges: [1001010000, 1001
019999], provider "restricted": Forbidden: not usable by user or serviceaccount, provider "nonroot-v2": Forbidden: not usable by user or serviceaccount, pr
ovider "nonroot": Forbidden: not usable by user or serviceaccount, provider "pcap-dedicated-admins": Forbidden: not usable by user or serviceaccount, provi
der "hostmount-anyuid": Forbidden: not usable by user or serviceaccount, provider "machine-api-termination-handler": Forbidden: not usable by user or servi
ceaccount, provider "hostnetwork-v2": Forbidden: not usable by user or serviceaccount, provider "hostnetwork": Forbidden: not usable by user or serviceacco
unt, provider "hostaccess": Forbidden: not usable by user or serviceaccount, provider "splunkforwarder": Forbidden: not usable by user or serviceaccount, p
rovider "trident-node-linux": Forbidden: not usable by user or serviceaccount, provider "node-exporter": Forbidden: not usable by user or serviceaccount, p
rovider "privileged": Forbidden: not usable by user or serviceaccount]
[ec2-user@ip-10-49-11-132 astra]$
```

Corrigez l'erreur en modifiant le `runAsUser` et `fsGroup` champs dans `statefuleset.apps/postgresql` objet avec l'uid qui se trouve dans la sortie du `oc get project` comme indiqué ci-dessous.

```
[ec2-user@ip-10-49-11-132 astra]$ oc get project postgresql -o yaml | grep uid-range
    openshift.io/sa.scc.uid-range: 1001010000/10000
[ec2-user@ip-10-49-11-132 astra]$ oc edit -n postgresql statefulset.apps/postgresql
statefulset.apps/postgresql edited
[ec2-user@ip-10-49-11-132 astra]$
```

L'application postgresql doit être en cours d'exécution et utiliser des volumes persistants pris en charge par le stockage Amazon FSX pour NetApp ONTAP.

```
[ec2-user@ip-10-49-11-132 astra]$ oc get pods -n postgresql
NAME            READY   STATUS    RESTARTS   AGE
postgresql-0    1/1     Running   0          2m46s
[ec2-user@ip-10-49-11-132 astra]$
```

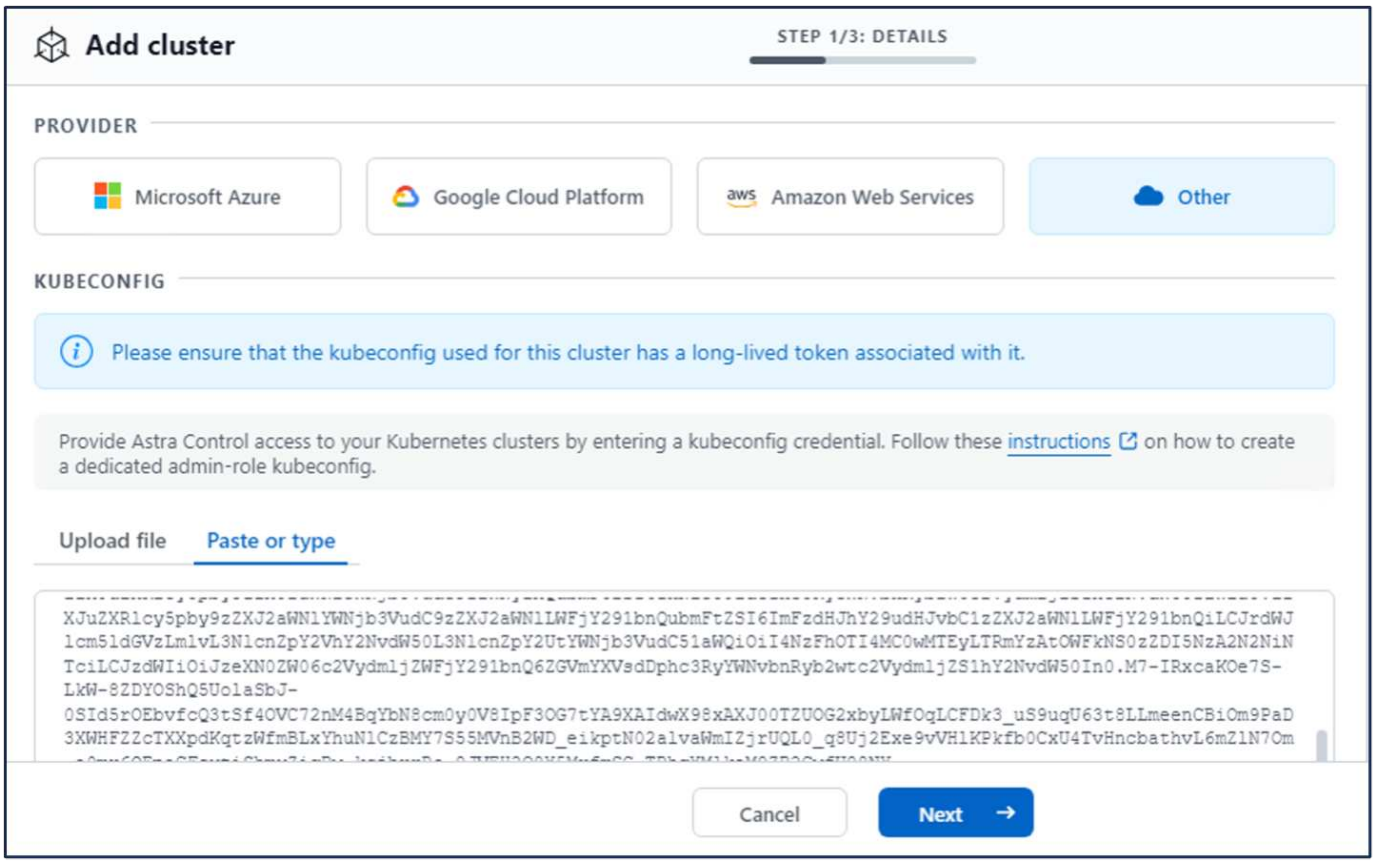
```
[ec2-user@ip-10-49-11-132 storage]$ kubectl get pvc -n postgresql
NAME                STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
data-postgresql-0   Bound    pvc-dd09524a-de75-4825-9424-03a9b91195ca   8Gi        RWO            ontap-nas      4m2s
[ec2-user@ip-10-49-11-132 storage]$
```

6. Créer une base de données et ajouter un enregistrement

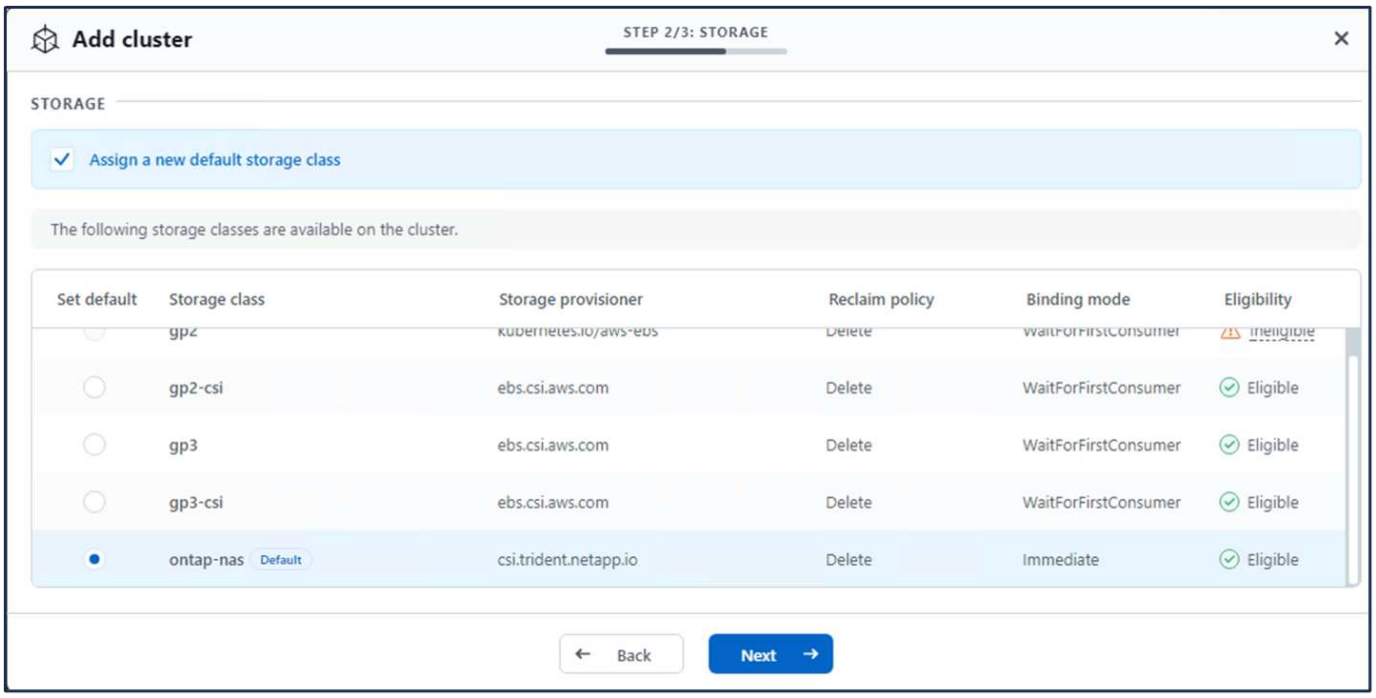
```
[ec2-user@ip-10-49-11-132 astra]$ export POSTGRES_PASSWORD=$(kubectl get secret --namespace postgresql postgresql -o jsonpath="
{.data.postgres-password}" | base64 -d)
[ec2-user@ip-10-49-11-132 astra]$ kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image
docker.io/bitnami/postgresql:16.2.0-debian-11-r1 --env="PGPASSWORD=$POSTGRES_PASSWORD" \
> --command -- psql --host postgresql -U postgres -d postgres -p 5432
Warning: would violate PodSecurity "restricted:v1.24": allowPrivilegeEscalation != false (container "postgresql-client" must se
t securityContext.allowPrivilegeEscalation=false), unrestricted capabilities (container "postgresql-client" must set securityCo
ntext.capabilities.drop=["ALL"]), runAsNonRoot != true (pod or container "postgresql-client" must set securityContext.runAsNonR
oot=true), seccompProfile (pod or container "postgresql-client" must set securityContext.seccompProfile.type to "RuntimeDefault
" or "Localhost")
If you don't see a command prompt, try pressing enter.

postgres=# CREATE DATABASE erp;
CREATE DATABASE
postgres=# \c erp
You are now connected to database "erp" as user "postgres".
erp=# CREATE TABLE PERSONS(ID INT PRIMARY KEY NOT NULL, FIRSTNAME TEXT NOT NULL, LASTNAME TEXT NOT NULL);
CREATE TABLE
erp=# INSERT INTO PERSONS VALUES(1,'John','Doe');
INSERT 0 1
erp=# \dt
          List of relations
 Schema |  Name   | Type  |  Owner
--------+---------+-------+----------
 public | persons | table | postgres
(1 row)

erp=# SELECT * FROM persons;
 id | firstname | lastname
----+-----------+----------
  1 | John      | Doe
(1 row)
```

7. Ajouter le cluster dans ACS

Connectez-vous à ACS. Sélectionnez cluster et cliquez sur Ajouter. Sélectionnez autre et téléchargez ou collez le fichier kubecconfig.



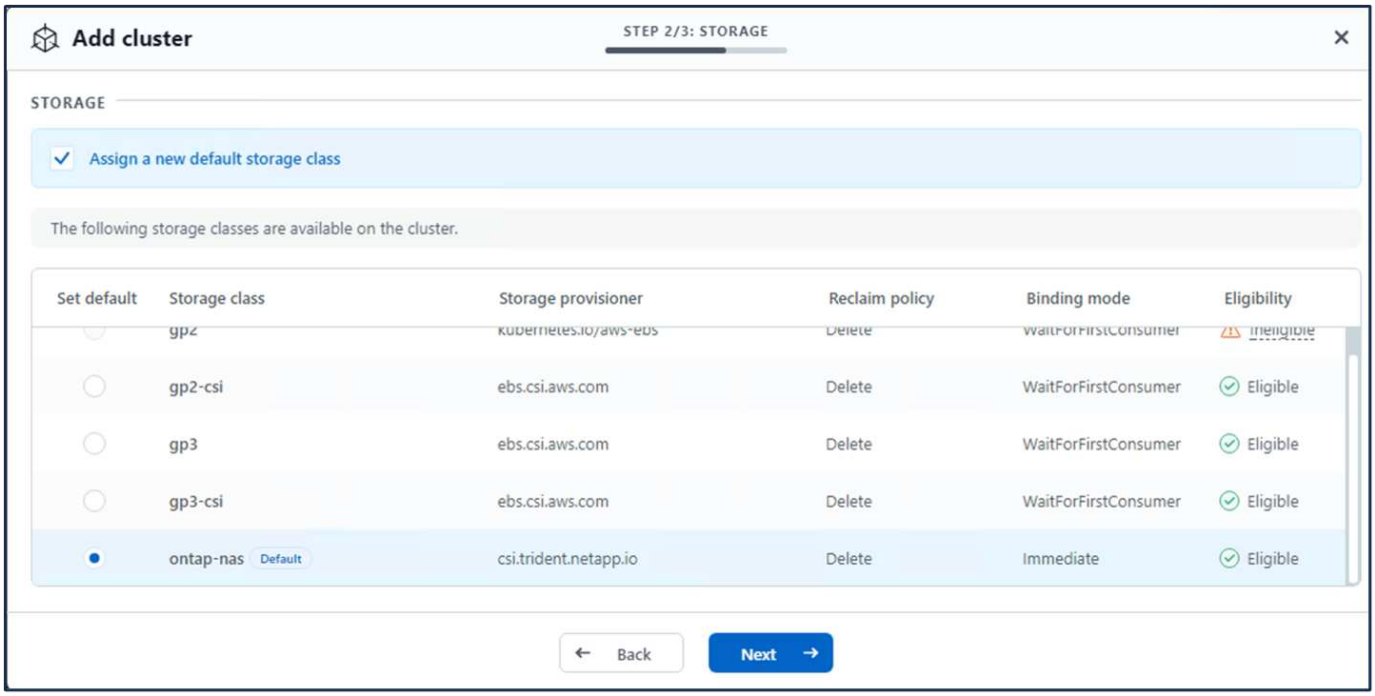
Cliquez sur **Suivant** et sélectionnez ontap-nas comme classe de stockage par défaut pour ACS. Cliquez sur **Suivant**, consultez les détails et **Ajouter** le cluster.



8. Définir l'application dans ACS

Définissez l'application postgresql dans ACS. Sur la page d'accueil, sélectionnez **applications**, **définir** et entrez les détails appropriés. Cliquez sur **Suivant** à plusieurs reprises, passez en revue les détails et cliquez

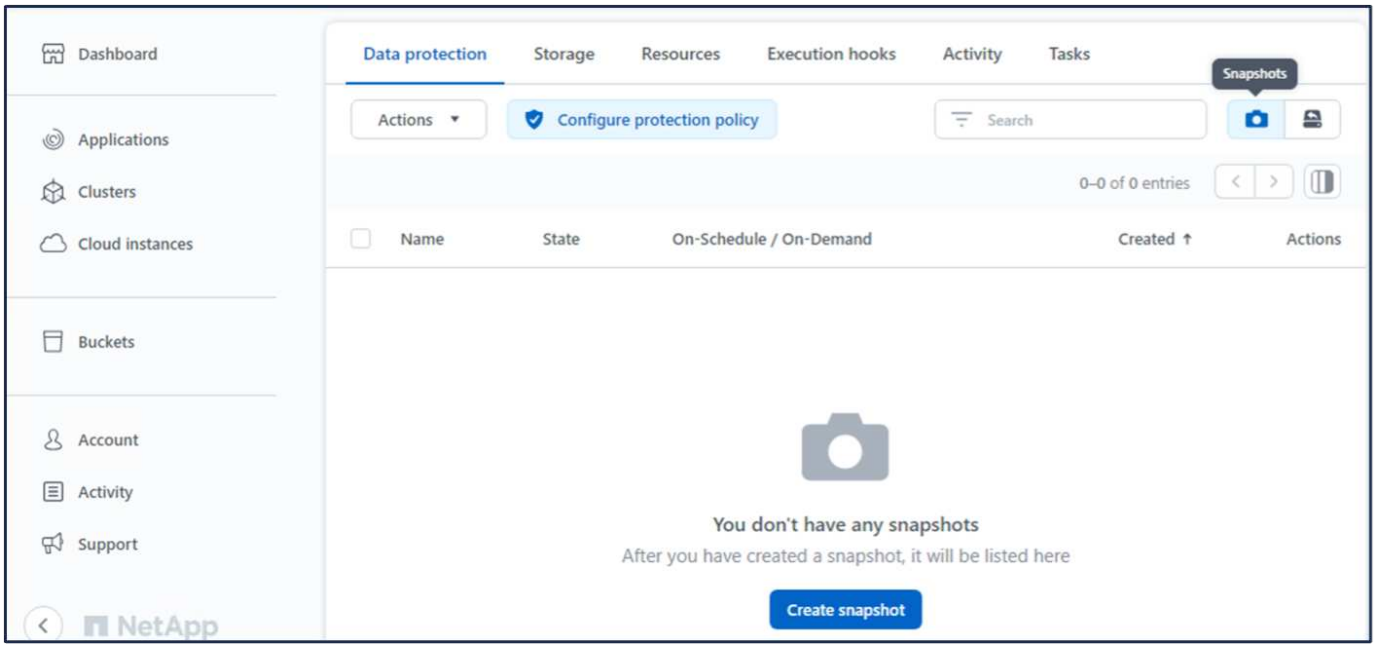
sur **définir**. L'application est ajoutée à ACS.

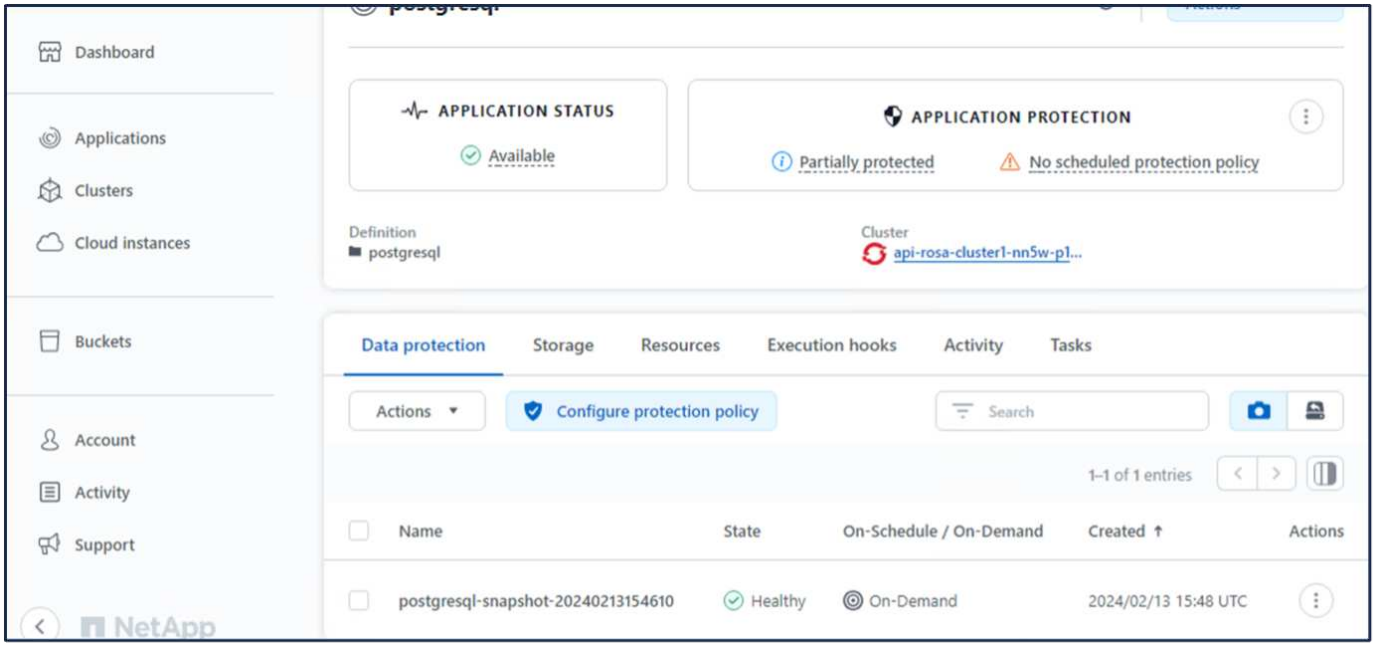


9. Créez un instantané à l'aide d'ACS

Il existe de nombreuses façons de créer un instantané dans ACS. Vous pouvez sélectionner l'application et créer un instantané à partir de la page qui affiche les détails de l'application. Vous pouvez cliquer sur Create snapshot pour créer un snapshot à la demande ou configurer une règle de protection.

Créez un instantané à la demande en cliquant simplement sur **Créer un instantané**, en fournissant un nom, en examinant les détails et en cliquant sur **instantané**. L'état de l'instantané passe à sain une fois l'opération terminée.





10. Supprimez la base de données dans l'application postgresql

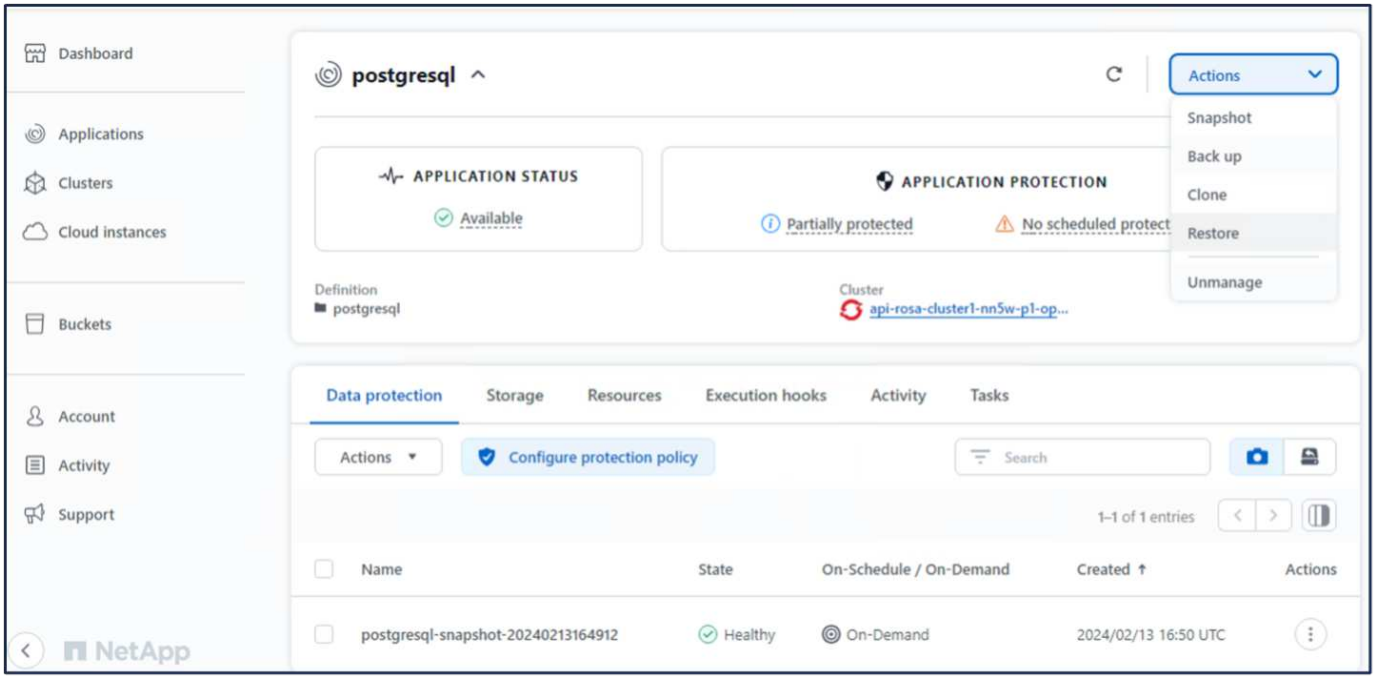
Reconnectez-vous à postgresql, répertoriez les bases de données disponibles, supprimez celle que vous avez créée précédemment et répertoriez à nouveau pour vous assurer que la base de données a été supprimée.

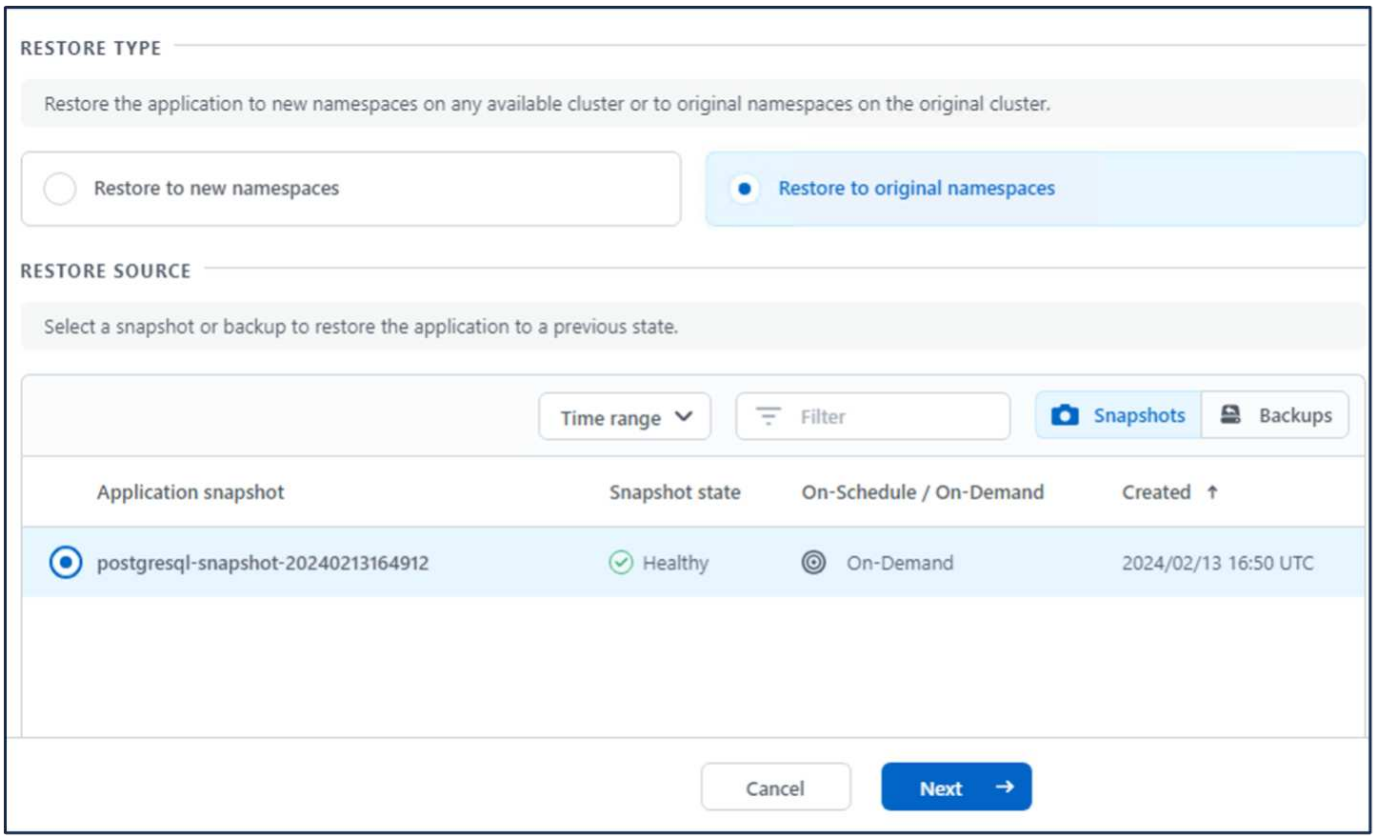
```
postgres=# \l
                                                    List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access priv
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+---------------
 erp       | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 postgres  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres
+          |          |          |                 |             |             |            |           | postgres=CTc/
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres
+          |          |          |                 |             |             |            |           | postgres=CTc/
(4 rows)

postgres=# DROP DATABASE erp;
DROP DATABASE
postgres=# \l
                                                    List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access priv
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+---------------
 postgres  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres
+          |          |          |                 |             |             |            |           | postgres=CTc/
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres
+          |          |          |                 |             |             |            |           | postgres=CTc/
(3 rows)
```

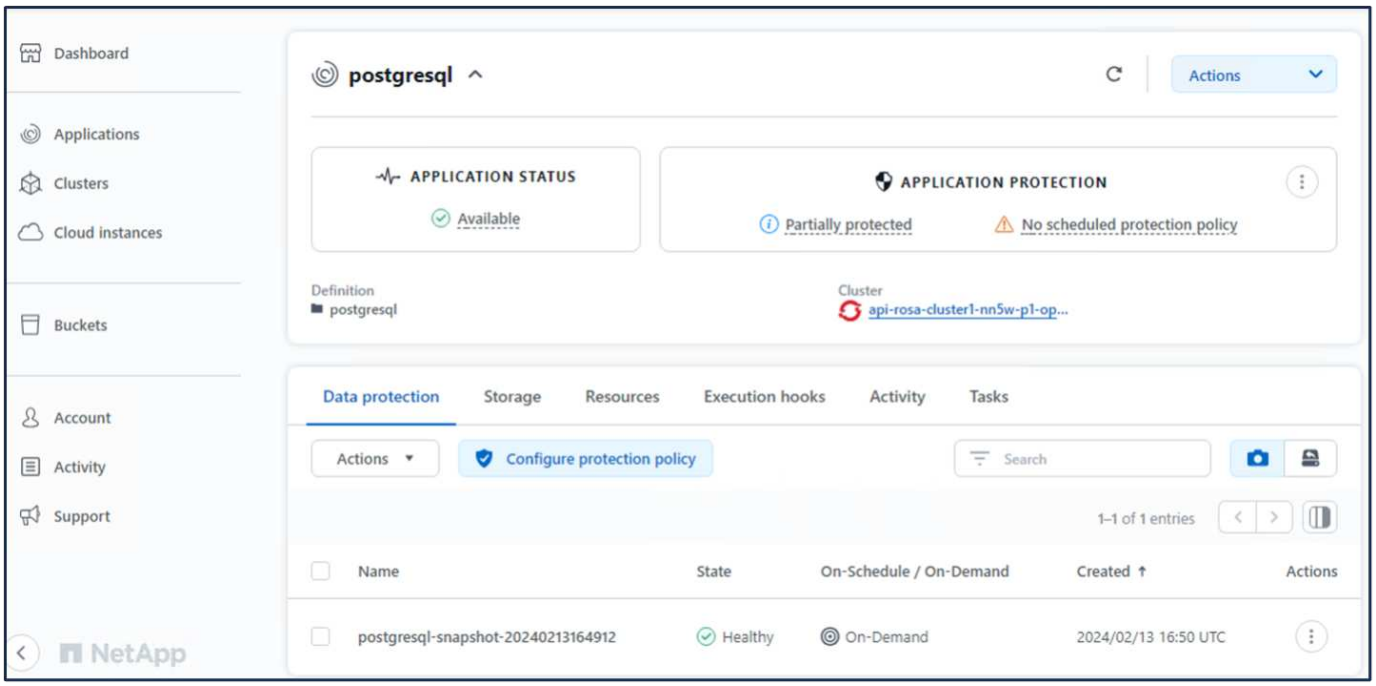
11. Restauration à partir d'un instantané à l'aide d'ACS

Pour restaurer l'application à partir d'un instantané, accédez à la page d'accueil de l'interface utilisateur ACS, sélectionnez l'application et sélectionnez Restaurer. Vous devez choisir un snapshot ou une sauvegarde à

partir de laquelle effectuer la restauration. (En général, plusieurs d'entre elles sont créées en fonction d'une règle que vous avez configurée). Faites les choix appropriés dans les deux écrans suivants, puis cliquez sur **Restaurer**. L'état de l'application passe de la restauration à disponible après sa restauration à partir de l'instantané.







12. Vérifiez que votre application a été restaurée à partir de l'instantané

Connectez-vous au client postgresql et vous devriez maintenant voir la table et l'enregistrement dans la table que vous aviez précédemment. C'est tout. En cliquant simplement sur un bouton, votre application a été restaurée à un état antérieur. C'est à ce niveau de simplicité que nous proposons à nos clients avec Astra Control.

```
[ec2-user@ip-10-49-11-132 ~]$ kubectl run postgresql-client --rm --tty -i --restart='Never' --namespace postgresql --image docker.io/bitnami/postgresql:16.
2.0-debian-11-r1 --env="PGPASSWORD=$POSTGRES_PASSWORD" --command -- psql --host postgresql -U postgres -d postgres -p 5432
Warning: would violate PodSecurity "restricted:v1.24": allowPrivilegeEscalation != false (container "postgresql-client" must set securityContext.allowPrivi
legeEscalation=false), unrestricted capabilities (container "postgresql-client" must set securityContext.capabilities.drop=["ALL"]), runAsNonRoot != true (
pod or container "postgresql-client" must set securityContext.runAsNonRoot=true), seccompProfile (pod or container "postgresql-client" must set securityCon
text.seccompProfile.type to "RuntimeDefault" or "Localhost")
If you don't see a command prompt, try pressing enter.

postgres=# \l
                                                    List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+-----------------------
 erp       | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 postgres  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
(4 rows)

postgres=# \c erp
You are now connected to database "erp" as user "postgres".
erp=# \dt
          List of relations
 Schema |  Name   | Type  |  Owner
--------+---------+-------+----------
 public | persons | table | postgres
(1 row)

erp=# SELECT * from PERSONS;
 id | firstname | lastname
----+-----------+----------
  1 | John      | Doe
(1 row)
```

Migration des données

Cette page présente les options de migration des données pour les workloads de conteneurs dans des clusters Red Hat OpenShift gérés à l'aide de FSX pour NetApp

ONTAP pour le stockage persistant.

Migration des données

Les services Red Hat OpenShift sur AWS et FSX pour NetApp ONTAP (FSxN) font partie de leur portefeuille de services par AWS. FSxN est disponible avec les options Single AZ ou Multi-AZ. L'option Multi-Az assure la protection des données contre les défaillances de zone de disponibilité. FSxN peut être intégré à Astra Trident pour fournir un stockage persistant aux applications sur les clusters ROSA.

Intégration de FSxN avec Trident à l'aide de Helm Chart

Intégration de clusters ROSA avec Amazon FSX for ONTAP

La migration des applications de conteneurs implique :

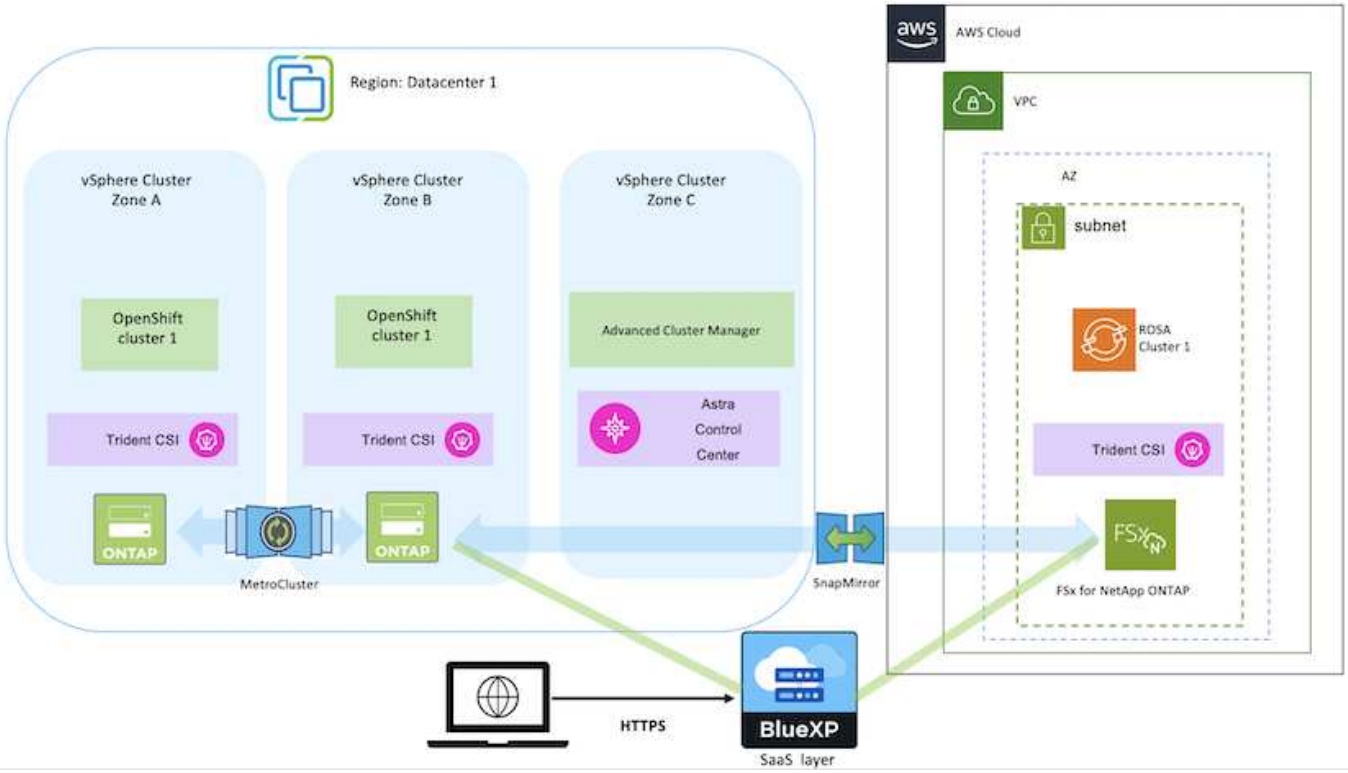
- Volumes persistants : cette opération peut être effectuée à l'aide de BlueXP. Une autre option consiste à utiliser Astra Control Center pour gérer les migrations d'applications de conteneurs d'un environnement sur site vers le cloud. L'automatisation peut être utilisée dans le même but.
- Métadonnées des applications : cette opération peut être réalisée à l'aide d'OpenShift GitOps (Argo CD).

Basculement et retour arrière des applications sur un cluster ROSA à l'aide de FSxN pour le stockage persistant

La vidéo suivante présente des scénarios de basculement et de retour arrière d'applications à l'aide de BlueXP et d'Argo CD.

Basculement et retour arrière des applications sur le cluster ROSA

Solution de protection et de migration des données pour les workloads de conteneurs OpenShift



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.