



Présentation de NetApp Astra Trident

NetApp Solutions

NetApp
April 26, 2024

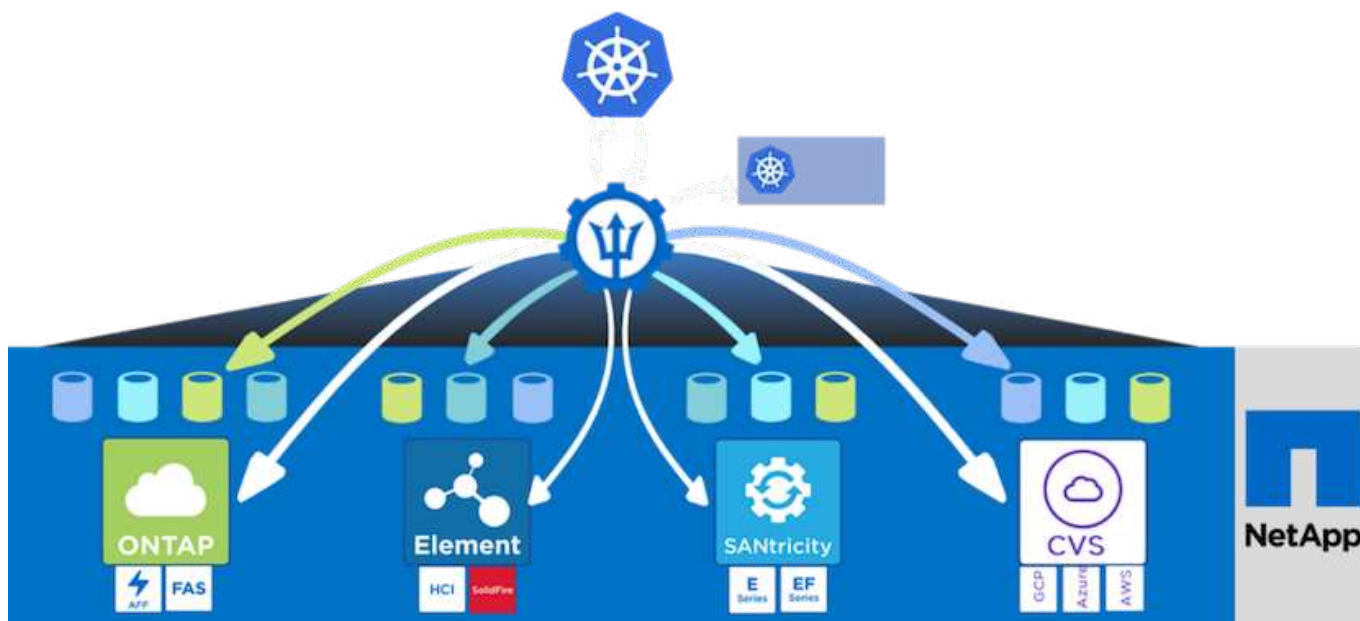
Sommaire

- Présentation d'Astra Trident 1
 - Téléchargez Astra Trident 1
 - Installer l'opérateur Trident avec Helm 3
 - Installez manuellement l'opérateur Trident 5
 - Préparez les nœuds workers pour le stockage 8
 - Création de systèmes back-end de stockage 13
 - Configuration NetApp ONTAP NFS 13
 - Configuration ONTAP iSCSI de NetApp 15
 - Configuration iSCSI de NetApp Element 18

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 laquelle 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%2Fs3%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%2Fs3%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,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXMgbm8KICAgICAgICBmaW5kX211bHRpcGF0aHMgbm8KfQoKYmxhY2tsaXN0X2V4Y2VwdGlvbnMgewogICAgICAgIHByb3BlcnR5ICIoU0NTSV9JREV0VF98SURfV1dOKSIKfQoKYmxhY2tsaXN0IHsKfQoK
          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: ""
```

- 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 référencée 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.template ./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.