Panoramica di NetApp Astra Trident

NetApp Solutions

NetApp
April 26, 2024

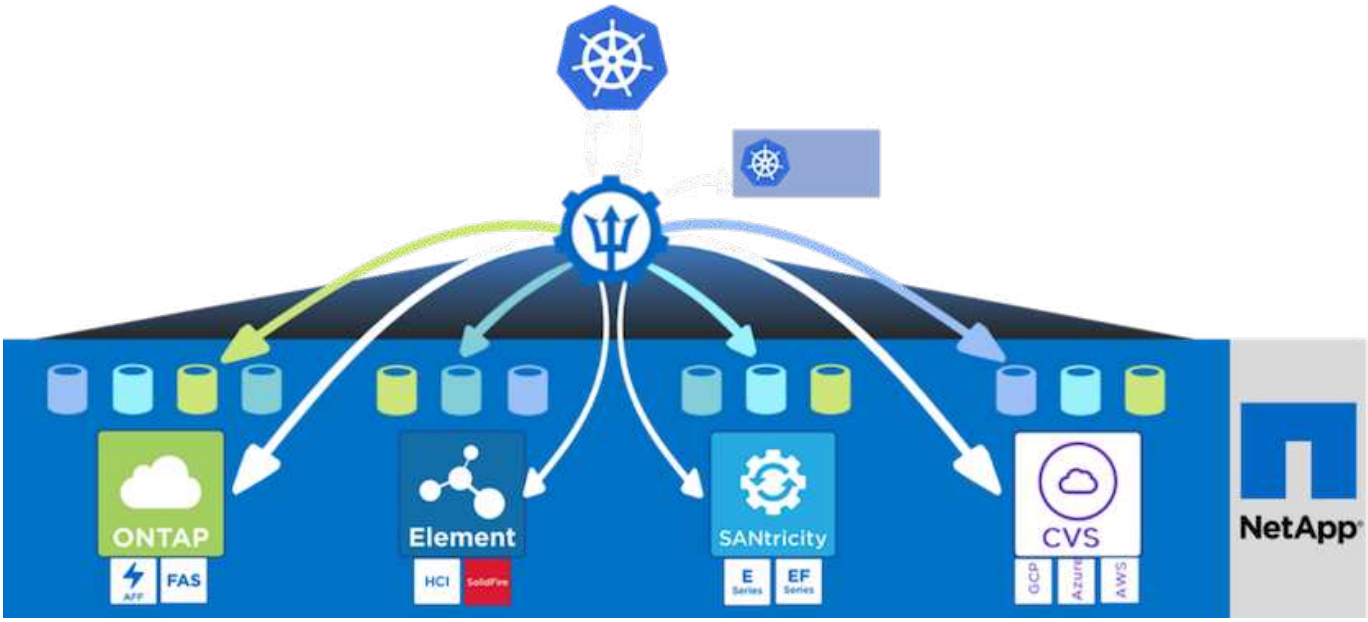
This PDF was generated from https://docs.netapp.com/it-it/netapp-solutions/containers/rh-os-n_trident_ontap_nfs.html on April 26, 2024. Always check docs.netapp.com for the latest.

Sommario

Panoramica di Astra Trident

Astra Trident è un orchestrator di storage open-source e completamente supportato per container e distribuzioni Kubernetes, incluso Red Hat OpenShift. Trident lavora con l'intero portfolio di storage NetApp, inclusi i sistemi storage NetApp ONTAP ed Element, e supporta anche connessioni NFS e iSCSI. Trident accelera il workflow DevOps consentendo agli utenti finali di eseguire il provisioning e gestire lo storage dai sistemi storage NetApp senza richiedere l'intervento di un amministratore dello storage.

Un amministratore può configurare una serie di backend di storage in base alle esigenze di progetto e ai modelli di sistemi di storage che consentono funzionalità di storage avanzate, tra cui compressione, tipi di dischi specifici o livelli di QoS che garantiscono un certo livello di performance. Una volta definiti, questi backend possono essere utilizzati dagli sviluppatori nei loro progetti per creare dichiarazioni di volume persistenti (PVC) e per collegare storage persistente ai propri container on-demand.



Astra Trident ha un rapido ciclo di sviluppo e, proprio come Kubernetes, viene rilasciato quattro volte all'anno.

L'ultima versione di Astra Trident è la 22.01 rilasciata a gennaio 2022. Matrice di supporto per quale versione di Trident è stata testata con la quale è possibile trovare la distribuzione Kubernetes "qui".

A partire dalla versione 20.04, l'impostazione di Trident viene eseguita dall'operatore Trident. L'operatore semplifica le implementazioni su larga scala e fornisce supporto aggiuntivo, inclusa la riparazione automatica dei pod implementati nell'installazione di Trident.

Con la versione 21.01, è stato reso disponibile un grafico Helm per facilitare l'installazione dell'operatore Trident.

Scarica Astra Trident

Per installare Trident sul cluster di utenti implementato ed eseguire il provisioning di un volume persistente, attenersi alla seguente procedura:

1. Scaricare l'archivio di installazione sulla workstation di amministrazione ed estrarre il contenuto. La versione corrente di Trident è la 22.01, che può essere scaricata "qui".

```
[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. Estrarre l'installazione di Trident dal bundle scaricato.

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

Installare l'operatore Trident con Helm

1. Innanzitutto, impostare la posizione del cluster utente `kubeconfig` File come variabile di ambiente in modo da non doverla fare riferimento, perché Trident non ha alcuna opzione per passare questo file.

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

2. Eseguire il comando Helm per installare l'operatore Trident dal tarball nella directory helm durante la creazione dello spazio dei nomi Trident nel cluster di utenti.

```
[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. È possibile verificare che Trident sia installato correttamente controllando i pod in esecuzione nello spazio dei nomi o utilizzando il binario tridentctl per controllare la versione installata.

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-5z45l                   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          |
+----------------+----------------+
```

In alcuni casi, gli ambienti dei clienti potrebbero richiedere la personalizzazione dell'implementazione di Trident. In questi casi, è anche possibile installare manualmente l'operatore Trident e aggiornare i manifesti inclusi per personalizzare l'implementazione.

Installare manualmente l'operatore Trident

1. Innanzitutto, impostare la posizione del cluster utente `kubeconfig` File come variabile di ambiente in modo da non doverla fare riferimento, perché Trident non ha alcuna opzione per passare questo file.

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

2. Il `trident-installer` la directory contiene i manifesti per la definizione di tutte le risorse richieste. Utilizzando i manifesti appropriati, creare `TridentOrchestrator` definizione personalizzata delle risorse.

```
[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. Se non ne esiste uno, creare uno spazio dei nomi Trident nel cluster utilizzando il manifesto fornito.

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

4. Creare le risorse necessarie per l'implementazione dell'operatore Trident, ad esempio un

`ServiceAccount` per l'operatore, un `ClusterRole` e. `ClusterRoleBinding` al `ServiceAccount`, un dedicato `PodSecurityPolicy`o l'operatore stesso.

```
[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. È possibile controllare lo stato dell'operatore dopo l'implementazione con i seguenti comandi:

```
[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. Con l'implementazione dell'operatore, ora possiamo utilizzarlo per installare Trident. Per eseguire questa operazione, è necessario creare 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:      kubectl-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. È possibile verificare che Trident sia installato correttamente controllando i pod in esecuzione nello spazio dei nomi o utilizzando il binario tridentctl per controllare la versione installata.

```
[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          |
+----------------+----------------+
```

Preparare i nodi di lavoro per lo storage

NFS

La maggior parte delle distribuzioni Kubernetes viene fornita con i pacchetti e le utility per montare i backend NFS installati di default, incluso Red Hat OpenShift.

Tuttavia, per NFSv3, non esiste alcun meccanismo per negoziare la concorrenza tra il client e il server.

Pertanto, il numero massimo di voci della tabella degli slot sunrpc lato client deve essere sincronizzato manualmente con il valore supportato sul server per garantire le migliori prestazioni per la connessione NFS senza che il server debba ridurre le dimensioni della finestra della connessione.

Per ONTAP, il numero massimo supportato di voci della tabella degli slot sunrpc è 128, ovvero ONTAP può gestire 128 richieste NFS simultanee alla volta. Tuttavia, per impostazione predefinita, Red Hat CoreOS/Red Hat Enterprise Linux ha un massimo di 65,536 voci della tabella degli slot sunrpc per connessione. È necessario impostare questo valore su 128 e questo può essere fatto usando Machine Config Operator (MCO) in OpenShift.

Per modificare il numero massimo di voci della tabella degli slot sunrpc nei nodi di lavoro OpenShift, attenersi alla seguente procedura:

1. Accedere alla console Web di OCP e selezionare Compute > Machine Configs (calcolo > configurazioni macchina). Fare clic su Create Machine Config. Copiare e incollare il file YAML e fare clic su Create (Crea).

```
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,b3B0aW9ucyBzdW5ycGMgdGNwX21heF9zbG90X3RhYmxlX2VudHJpZXM9MTI4Cg=
=
          filesystem: root
          mode: 420
          path: /etc/modprobe.d/sunrpc.conf
```

2. Dopo aver creato l'MCO, la configurazione deve essere applicata a tutti i nodi di lavoro e riavviata uno alla volta. L'intero processo richiede da 20 a 30 minuti circa. Verificare se la configurazione del computer viene applicata utilizzando `oc get mcp` e assicurarsi che il pool di configurazione del computer per i lavoratori sia aggiornato.

```
[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

Per preparare i nodi di lavoro per consentire la mappatura dei volumi di storage a blocchi tramite il protocollo iSCSI, è necessario installare i pacchetti necessari per supportare tale funzionalità.

In Red Hat OpenShift, questo viene gestito applicando un MCO (Machine Config Operator) al cluster dopo averlo implementato.

Per configurare i nodi di lavoro per l'esecuzione dei servizi iSCSI, attenersi alla seguente procedura:

1. Accedere alla console Web di OCP e selezionare Compute > Machine Configs (calcolo > configurazioni macchina). Fare clic su Create Machine Config. Copiare e incollare il file YAML e fare clic su Create (Crea).

   Quando non si utilizza il multipathing:

   ```
   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: ""
   ```

   Quando si utilizza il multipathing:

```
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,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXMgbm8KICAgICA
gICBmaW5kX211bHRpcGF0aHMgbm8KfQoKYmxhY2tsaXN0X2V4Y2VwdGlvbnMgewogICAgICA
gIHByb3BlcnR5ICIoU0NTSV9JREVOVF98SURfV1dOKSIKfQoKYmxhY2tsaXN0IHsKfQoK
          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. Una volta creata la configurazione, sono necessari circa 20 - 30 minuti per applicarla ai nodi di lavoro e ricaricarla. Verificare se la configurazione del computer viene applicata utilizzando `oc get mcp` e assicurarsi che il pool di configurazione del computer per i lavoratori sia aggiornato. È inoltre possibile accedere ai nodi di lavoro per confermare che il servizio iscsid è in esecuzione (e il servizio multipath è in esecuzione se si utilizza il multipathing).

```
[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
```

È inoltre possibile confermare che MachineConfig sia stato applicato correttamente e che i servizi siano stati avviati come previsto eseguendo il `oc debug` con i flag appropriati.

Creazione di backend per il sistema storage

Dopo aver completato l'installazione di Astra Trident Operator, è necessario configurare il backend per la

piattaforma di storage NetApp specifica in uso. Seguire i collegamenti riportati di seguito per continuare l'installazione e la configurazione di Astra Trident.

- "NetApp ONTAP NFS"
- "ISCSI NetApp ONTAP"
- "ISCSI NetApp Element"

Configurazione NFS di NetApp ONTAP

Per consentire l'integrazione di Trident con il sistema storage NetApp ONTAP, è necessario creare un backend che consenta la comunicazione con il sistema storage.

1. Nell'archivio di installazione scaricato in sono disponibili file backend di esempio `sample-input` gerarchia di cartelle. Per i sistemi NetApp ONTAP che servono NFS, copiare il `backend-ontap-nas.json` nella directory di lavoro e modificare il file.

```
[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. Modificare il backendName, managementLIF, dataLIF, svm, nome utente, e password in questo file.

```
{
    "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"
}
```

> È consigliabile definire il valore backendName personalizzato come combinazione di storageDriverName e dataLIF che fornisce NFS per una facile identificazione.

3. Una volta creato questo file di back-end, eseguire il comando seguente per creare il primo backend.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-nas.json
+------------------------+----------------
+--------------------------------------+--------+---------+
|          NAME          | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+------------------------+----------------
+--------------------------------------+--------+---------+
| ontap-nas+10.61.181.221 | ontap-nas      | be7a619d-c81d-445c-b80c-
5c87a73c5b1e | online |       0 |
+------------------------+----------------
+--------------------------------------+--------+---------+
```

4. Una volta creato il backend, è necessario creare una classe di storage. Come per il backend, esiste un file di esempio della classe di storage che può essere modificato per l'ambiente disponibile nella cartella di input di esempio. Copiarlo nella directory di lavoro e apportare le modifiche necessarie per riflettere il backend creato.

```
[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. L'unica modifica che deve essere apportata a questo file è definire `backendType` valore al nome del driver di storage dal backend appena creato. Annotare anche il valore del campo nome, a cui si deve fare riferimento in un passaggio successivo.

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

> Esiste un campo opzionale chiamato `fsType` definito in questo file. Questa riga può essere eliminata nei backend NFS.

6. Eseguire `oc` per creare la classe di storage.

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

7. Una volta creata la classe di storage, è necessario creare la prima dichiarazione di volume persistente (PVC). C'è un esempio `pvc-basic.yaml` file che può essere utilizzato per eseguire questa azione, disponibile anche in input di esempio.

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

8. L'unica modifica che deve essere apportata a questo file è garantire che il `storageClassName` il campo corrisponde a quello appena creato. La definizione PVC può essere ulteriormente personalizzata in base alle esigenze del carico di lavoro da fornire.

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

9. Creare il PVC emettendo il `oc` comando. La creazione può richiedere del tempo a seconda delle dimensioni del volume di backup da creare, in modo da poter guardare il processo mentre viene completato.

```
[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
```

Configurazione iSCSI di NetApp ONTAP

Per consentire l'integrazione di Trident con il sistema storage NetApp ONTAP, è necessario creare un backend che consenta la comunicazione con il sistema storage.

1. Nell'archivio di installazione scaricato in sono disponibili file backend di esempio `sample-input` gerarchia di cartelle. Per i sistemi NetApp ONTAP che utilizzano iSCSI, copiare il `backend-ontap-san.json` nella

directory di lavoro e modificare il file.

```
[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. Modificare i valori di gestione LIF, dataLIF, svm, nome utente e password in questo file.

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

3. Una volta creato questo file di back-end, eseguire il comando seguente per creare il primo 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. Una volta creato il backend, è necessario creare una classe di storage. Come per il backend, esiste un file di esempio della classe di storage che può essere modificato per l'ambiente disponibile nella cartella di input di esempio. Copiarlo nella directory di lavoro e apportare le modifiche necessarie per riflettere il backend creato.

```
[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. L'unica modifica che deve essere apportata a questo file è definire `backendType` valore al nome del driver

di storage dal backend appena creato. Annotare anche il valore del campo nome, a cui si deve fare riferimento in un passaggio successivo.

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

> Esiste un campo opzionale chiamato `fsType` definito in questo file. Nei backend iSCSI, questo valore può essere impostato su un tipo di filesystem Linux specifico (XFS, ext4, ecc.) o può essere cancellato per consentire a OpenShift di decidere quale filesystem usare.

6. Eseguire `oc` per creare la classe di storage.

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

7. Una volta creata la classe di storage, è necessario creare la prima dichiarazione di volume persistente (PVC). C'è un esempio `pvc-basic.yaml` file che può essere utilizzato per eseguire questa azione, disponibile anche in input di esempio.

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

8. L'unica modifica che deve essere apportata a questo file è garantire che il `storageClassName` il campo corrisponde a quello appena creato. La definizione PVC può essere ulteriormente personalizzata in base alle esigenze del carico di lavoro da fornire.

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

9. Creare il PVC emettendo il `oc` comando. La creazione può richiedere del tempo a seconda delle dimensioni del volume di backup da creare, in modo da poter guardare il processo mentre viene completato.

```
[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
```

Configurazione iSCSI NetApp Element

Per abilitare l'integrazione di Trident con il sistema storage NetApp Element, è necessario creare un backend che consenta la comunicazione con il sistema storage utilizzando il protocollo iSCSI.

1. Nell'archivio di installazione scaricato in sono disponibili file backend di esempio `sample-input` gerarchia di cartelle. Per i sistemi NetApp Element che utilizzano iSCSI, copiare `backend-solidfire.json` nella directory di lavoro e modificare il file.

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

   a. Modificare i valori di utente, password e MVIP su `EndPoint` linea.

   b. Modificare il `SVIP` valore.

```
 {
    "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. Una volta creato questo file back-end, eseguire il seguente comando per creare il primo backend.

```
[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. Una volta creato il backend, è necessario creare una classe di storage. Come per il backend, esiste un file di esempio della classe di storage che può essere modificato per l'ambiente disponibile nella cartella di input di esempio. Copiarlo nella directory di lavoro e apportare le modifiche necessarie per riflettere il backend creato.

```
[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
```

4. L'unica modifica che deve essere apportata a questo file è definire `backendType` valore al nome del driver di storage dal backend appena creato. Annotare anche il valore del campo nome, a cui si deve fare riferimento in un passaggio successivo.

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

> Esiste un campo opzionale chiamato `fsType` definito in questo file. Nei backend iSCSI, questo valore può essere impostato su un tipo di filesystem Linux specifico (XFS, ext4 e così via), oppure può essere cancellato per consentire a OpenShift di decidere quale filesystem usare.

5. Eseguire `oc` per creare la classe di storage.

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

6. Una volta creata la classe di storage, è necessario creare la prima dichiarazione di volume persistente (PVC). C'è un esempio `pvc-basic.yaml` file che può essere utilizzato per eseguire questa azione, disponibile anche in input di esempio.

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

7. L'unica modifica che deve essere apportata a questo file è garantire che il `storageClassName` il campo corrisponde a quello appena creato. La definizione PVC può essere ulteriormente personalizzata in base alle esigenze del carico di lavoro da fornire.

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

8. Creare il PVC emettendo il `oc` comando. La creazione può richiedere del tempo a seconda delle dimensioni del volume di backup da creare, in modo da poter guardare il processo mentre viene completato.

```
[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-3445b5cc-df24-453d-a1e6-b484e874349d   1Gi
RWO            basic-csi      5s
```

Informazioni sul copyright

Copyright © 2024 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIABILITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEQUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE: l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

Informazioni sul marchio commerciale

NETAPP, il logo NETAPP e i marchi elencati alla pagina http://www.netapp.com/TM sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.