Gestire e monitorare Trident

Trident

NetApp
January 15, 2026

Sommario

Gestire e monitorare Trident

Trident potenziato

Trident potenziato

A partire dalla versione 24.02, Trident segue una cadenza di rilascio di quattro mesi, rilasciando tre versioni principali ogni anno solare. Ogni nuova versione si basa sulle versioni precedenti e offre nuove funzionalità, miglioramenti delle prestazioni, correzioni di bug e miglioramenti. Ti invitiamo ad effettuare l'aggiornamento almeno una volta all'anno per sfruttare le nuove funzionalità di Trident.

Considerazioni prima dell'aggiornamento

Quando si esegue l'aggiornamento all'ultima versione di Trident, tenere presente quanto segue:

- Dovrebbe essere installata una sola istanza Trident in tutti gli spazi dei nomi di un determinato cluster Kubernetes.
- Trident 23.07 e versioni successive richiedono snapshot del volume v1 e non supportano più snapshot alpha o beta.
- Se hai creato il Cloud Volumes Service per Google Cloud in"Tipo di servizio CVS" , è necessario aggiornare la configurazione del backend per utilizzare `standardsw` O `zoneredundantstandardsw` livello di servizio durante l'aggiornamento da Trident 23.01. Mancato aggiornamento del `serviceLevel` nel backend potrebbe causare il fallimento dei volumi. Fare riferimento a "Esempi di tipi di servizio CVS" per i dettagli.
- Quando si esegue l'aggiornamento, è importante fornire `parameter.fsType` In `StorageClasses` utilizzato da Trident. Puoi eliminare e ricreare `StorageClasses` senza interrompere i volumi preesistenti.
  - Questo è un **requisito** per l'applicazione "contesti di sicurezza" per volumi SAN.
  - La directory https://github.com/NetApp/trident/tree/master/trident-installer/sample-input [sample input^] contiene esempi, come `storage-class-basic.yaml.templ` e collegamento: `storage-class-bronze-default.yaml` .
  - Per maggiori informazioni, fare riferimento a"Problemi noti" .

Passaggio 1: seleziona una versione

Le versioni Trident seguono un sistema basato sulla data `YY.MM` convenzione di denominazione, dove "YY" rappresenta le ultime due cifre dell'anno e "MM" è il mese. Le versioni Dot seguono un `YY.MM.X` convenzione, dove "X" è il livello di patch. Selezionerai la versione a cui effettuare l'aggiornamento in base alla versione da cui stai effettuando l'aggiornamento.

- È possibile eseguire un aggiornamento diretto a qualsiasi versione di destinazione che si trovi entro una finestra temporale di quattro versioni dalla versione installata. Ad esempio, è possibile effettuare l'aggiornamento direttamente dalla versione 24.06 (o da qualsiasi versione 24.06 dot) alla versione 25.06.
- Se si esegue l'aggiornamento da una versione al di fuori della finestra delle quattro versioni, eseguire un aggiornamento in più fasi. Utilizzare le istruzioni di aggiornamento per"versione precedente" stai eseguendo l'aggiornamento alla versione più recente che rientra nella finestra temporale delle quattro versioni. Ad esempio, se stai utilizzando la versione 23.07 e vuoi eseguire l'aggiornamento alla versione 25.06:

a. Primo aggiornamento dal 23.07 al 24.06.

b. Quindi aggiorna dalla versione 24.06 alla versione 25.06.

> Quando si esegue l'aggiornamento tramite l'operatore Trident su OpenShift Container Platform, è necessario eseguire l'aggiornamento a Trident 21.01.1 o versione successiva. L'operatore Trident rilasciato con la versione 21.01.0 contiene un problema noto che è stato risolto nella versione 21.01.1. Per maggiori dettagli fare riferimento al "dettagli del problema su GitHub" .

Passaggio 2: determinare il metodo di installazione originale

Per determinare quale versione hai utilizzato per installare originariamente Trident:

1. Utilizzo `kubectl get pods -n trident` per esaminare i baccelli.
   - Se non è presente alcun pod operatore, Trident è stato installato utilizzando `tridentctl` .
   - Se è presente un pod operatore, Trident è stato installato utilizzando l'operatore Trident manualmente o tramite Helm.
2. Se è presente un pod operatore, utilizzare `kubectl describe torc` per determinare se Trident è stato installato tramite Helm.
   - Se è presente un'etichetta Helm, Trident è stato installato tramite Helm.
   - Se non è presente alcuna etichetta Helm, Trident è stato installato manualmente utilizzando l'operatore Trident .

Passaggio 3: seleziona un metodo di aggiornamento

In genere, dovresti eseguire l'aggiornamento utilizzando lo stesso metodo utilizzato per l'installazione iniziale, tuttavia puoi"spostarsi tra i metodi di installazione" . Ci sono due opzioni per potenziare Trident.

- "Aggiorna utilizzando l'operatore Trident"

> Ti suggeriamo di rivedere"Comprendere il flusso di lavoro di aggiornamento dell'operatore" prima di effettuare l'aggiornamento con l'operatore.

*

Aggiorna con l'operatore

Comprendere il flusso di lavoro di aggiornamento dell'operatore

Prima di utilizzare l'operatore Trident per aggiornare Trident, è necessario comprendere i processi in background che si verificano durante l'aggiornamento. Ciò include modifiche al controller Trident , al controller Pod e ai node Pod, nonché al node DaemonSet che consentono aggiornamenti continui.

Gestione dell'aggiornamento dell'operatore Trident

Uno dei tanti"vantaggi dell'utilizzo dell'operatore Trident" per installare e aggiornare Trident è la gestione automatica degli oggetti Trident e Kubernetes senza interrompere i volumi montati esistenti. In questo modo, Trident può supportare gli aggiornamenti senza tempi di inattività, oppure"*aggiornamenti continui*" . In particolare, l'operatore Trident comunica con il cluster Kubernetes per:

- Eliminare e ricreare la distribuzione Trident Controller e il nodo DaemonSet.
- Sostituisci i Trident Controller Pod e i Trident Node Pod con nuove versioni.
  - Se un nodo non viene aggiornato, ciò non impedisce l'aggiornamento dei nodi rimanenti.
  - Solo i nodi con un Trident Node Pod in esecuzione possono montare volumi.

Per ulteriori informazioni sull'architettura Trident sul cluster Kubernetes, fare riferimento a"Architettura Trident" .

Flusso di lavoro di aggiornamento dell'operatore

Quando si avvia un aggiornamento utilizzando l'operatore Trident :

1. L'operatore * Trident *:
   a. Rileva la versione di Trident attualmente installata (versione *n*).
   b. Aggiorna tutti gli oggetti Kubernetes, inclusi CRD, RBAC e Trident SVC.
   c. Elimina la distribuzione Trident Controller per la versione *n*.
   d. Crea la distribuzione Trident Controller per la versione *n+1*.
2. **Kubernetes** crea il Trident Controller Pod per *n+1*.
3. L'operatore * Trident *:
   a. Elimina il Trident Node DaemonSet per *n*. L'operatore non attende la terminazione del Node Pod.
   b. Crea il Trident Node Daemonset per *n+1*.
4. **Kubernetes** crea Trident Node Pod sui nodi che non eseguono Trident Node Pod *n*. Ciò garantisce che non ci sia mai più di un Trident Node Pod, di qualsiasi versione, su un nodo.

Aggiorna un'installazione Trident utilizzando l'operatore Trident o Helm

È possibile aggiornare Trident utilizzando l'operatore Trident manualmente o tramite Helm. È possibile effettuare l'aggiornamento da un'installazione dell'operatore Trident a un'altra installazione dell'operatore Trident o … `tridentctl` installazione su una versione dell'operatore Trident . Revisione"Seleziona un metodo di aggiornamento" prima di aggiornare l'installazione di un operatore Trident .

Aggiornare un'installazione manuale

È possibile eseguire l'aggiornamento da un'installazione dell'operatore Trident con ambito cluster a un'altra installazione dell'operatore Trident con ambito cluster. Tutte le versioni Trident utilizzano un operatore con ambito cluster.

Per eseguire l'aggiornamento da Trident installato utilizzando l'operatore con ambito namespace (versioni dalla 20.07 alla 20.10), utilizzare le istruzioni di aggiornamento per"la tua versione installata" del Trident.

Informazioni su questo compito

Trident fornisce un file bundle che puoi utilizzare per installare l'operatore e creare oggetti associati per la tua versione di Kubernetes.

- Per i cluster che eseguono Kubernetes 1.24, utilizzare"bundle_pre_1_25.yaml" .
- Per i cluster che eseguono Kubernetes 1.25 o versioni successive, utilizzare"bundle_post_1_25.yaml" .

Prima di iniziare

Assicurati di utilizzare un cluster Kubernetes in esecuzione"una versione di Kubernetes supportata" .

Passi

1. Verifica la tua versione Trident :

```
./tridentctl -n trident version
```

2. Aggiorna il `operator.yaml` , `tridentorchestrator_cr.yaml` , E `post_1_25_bundle.yaml` con il registro e i percorsi immagine per la versione a cui si sta effettuando l'aggiornamento (ad esempio 25.06) e il segreto corretto.
3. Eliminare l'operatore Trident utilizzato per installare l'istanza Trident corrente. Ad esempio, se si esegue l'aggiornamento dalla versione 25.02, eseguire il seguente comando:

```
kubectl delete -f 25.02.0/trident-installer/deploy/<bundle.yaml> -n
trident
```

4. Se hai personalizzato l'installazione iniziale utilizzando `TridentOrchestrator` attributi, è possibile modificare il `TridentOrchestrator` oggetto per modificare i parametri di installazione. Ciò potrebbe includere modifiche apportate per specificare registri di immagini Trident e CSI speculari per la modalità offline, abilitare registri di debug o specificare segreti di estrazione delle immagini.
5. Installa Trident utilizzando il file YAML del bundle corretto per il tuo ambiente, dove *<bundle.yaml>* è `bundle_pre_1_25.yaml` O `bundle_post_1_25.yaml` in base alla versione di Kubernetes. Ad esempio, se si installa Trident 25.06.0, eseguire il seguente comando:

```
kubectl create -f 25.06.0/trident-installer/deploy/<bundle.yaml> -n
trident
```

6. Modifica la torcia del tridente per includere l'immagine 25.06.0.

Aggiornare un'installazione di Helm

È possibile aggiornare un'installazione Trident Helm.

> Quando si aggiorna un cluster Kubernetes da 1.24 a 1.25 o versione successiva su cui è installato Trident , è necessario aggiornare values.yaml per impostare `excludePodSecurityPolicy` A `true` o aggiungere `--set excludePodSecurityPolicy=true` al `helm upgrade` comando prima di poter aggiornare il cluster.

Se hai già aggiornato il tuo cluster Kubernetes dalla versione 1.24 alla versione 1.25 senza aggiornare Trident Helm, l'aggiornamento di Helm non riesce. Per completare l'aggiornamento del timone, è necessario eseguire questi passaggi come prerequisiti:

1. Installa il plugin helm-mapkubeapis da https://github.com/helm/helm-mapkubeapis .
2. Eseguire una prova di funzionamento della versione Trident nello spazio dei nomi in cui è installato Trident . Qui sono elencate le risorse che verranno ripulite.

```
helm mapkubeapis --dry-run trident  --namespace trident
```

3. Eseguire una corsa completa con il timone per effettuare la pulizia.

```
helm mapkubeapis trident  --namespace trident
```

Passi

1. Se tu"installato Trident tramite Helm" , puoi usare `helm upgrade trident netapp-trident/trident-operator --version 100.2506.0` per aggiornare in un unico passaggio. Se non hai aggiunto il repository Helm o non puoi utilizzarlo per l'aggiornamento:
   a. Scarica l'ultima versione Trident da"la sezione *Assets* su GitHub" .
   b. Utilizzare il `helm upgrade` comando dove `trident-operator-25.06.0.tgz` riflette la versione a cui si desidera effettuare l'aggiornamento.

```
helm upgrade <name> trident-operator-25.06.0.tgz
```

> Se si impostano opzioni personalizzate durante l'installazione iniziale (ad esempio specificando registri privati e speculari per le immagini Trident e CSI), aggiungere `helm upgrade` comando utilizzando `--set` per garantire che tali opzioni siano incluse nel comando di aggiornamento, altrimenti i valori verranno ripristinati ai valori predefiniti.

2. Correre `helm list` per verificare che sia la versione del grafico che quella dell'app siano state aggiornate. Correre `tridentctl logs` per rivedere eventuali messaggi di debug.

Aggiorna da un `tridentctl` installazione all'operatore Trident

È possibile effettuare l'aggiornamento all'ultima versione dell'operatore Trident da un `tridentctl` installazione. I backend e i PVC esistenti saranno automaticamente disponibili.

> Prima di passare da un metodo di installazione all'altro, rivedere"Spostamento tra i metodi di installazione" .

Passi

1. Scarica l'ultima versione Trident .

```
# Download the release required [25.06.0]
mkdir 25.06.0
cd 25.06.0
wget
https://github.com/NetApp/trident/releases/download/v25.06.0/trident-
installer-25.06.0.tar.gz
tar -xf trident-installer-25.06.0.tar.gz
cd trident-installer
```

2. Crea il `tridentorchestrator` CRD dal manifesto.

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

3. Distribuire l'operatore con ambito cluster nello stesso namespace.

```
kubectl create -f deploy/<bundle-name.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

#Examine the pods in the Trident namespace
NAME                                READY   STATUS    RESTARTS   AGE
trident-controller-79df798bdc-m79dc   6/6     Running   0          150d
trident-node-linux-xrst8              2/2     Running   0          150d
trident-operator-5574dbbc68-nthjv     1/1     Running   0          1m30s
```

4. Crea un `TridentOrchestrator` CR per l'installazione Trident.

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

kubectl create -f deploy/crds/tridentorchestrator_cr.yaml

#Examine the pods in the Trident namespace
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-79df798bdc-m79dc        6/6     Running   0          1m
trident-csi-xrst8                   2/2     Running   0          1m
trident-operator-5574dbbc68-nthjv   1/1     Running   0          5m41s
```

5. Conferma che Trident è stato aggiornato alla versione prevista.

```
kubectl describe torc trident | grep Message -A 3

Message:                Trident installed
Namespace:              trident
Status:                 Installed
Version:                v25.06.0
```

Aggiorna con tridentctl

È possibile aggiornare facilmente un'installazione Trident esistente utilizzando `tridentctl`.

Informazioni su questo compito

Disinstallare e reinstallare Trident equivale a eseguire un aggiornamento. Quando si disinstalla Trident, il Persistent Volume Claim (PVC) e il Persistent Volume (PV) utilizzati dalla distribuzione Trident non vengono eliminati. I PV già forniti rimarranno disponibili mentre Trident è offline e Trident fornirà volumi per tutti i PVC creati nel frattempo, dopo essere tornato online.

Prima di iniziare

Revisione"Seleziona un metodo di aggiornamento" prima di effettuare l'aggiornamento utilizzando `tridentctl`.

Passi

1. Eseguire il comando di disinstallazione in `tridentctl` per rimuovere tutte le risorse associate a Trident, ad eccezione dei CRD e degli oggetti correlati.

```
./tridentctl uninstall -n <namespace>
```

2. Reinstallare Trident. Fare riferimento a"Installa Trident usando tridentctl" .

> Non interrompere il processo di aggiornamento. Assicurarsi che il programma di installazione venga completato.

Gestisci Trident usando tridentctl

IL "Pacchetto di installazione Trident" include il `tridentctl` utilità della riga di comando per fornire un accesso semplice a Trident. Gli utenti Kubernetes con privilegi sufficienti possono utilizzarlo per installare Trident o gestire lo spazio dei nomi che contiene il pod Trident .

Comandi e flag globali

Puoi correre `tridentctl help` per ottenere un elenco dei comandi disponibili per `tridentctl` o aggiungere il `--help` flag a qualsiasi comando per ottenere un elenco di opzioni e flag per quel comando specifico.

```
tridentctl [command] [--optional-flag]
```

Il Trident `tridentctl` L'utilità supporta i seguenti comandi e flag globali.

Comandi

`create`
Aggiungi una risorsa a Trident.

`delete`
Rimuovi una o più risorse da Trident.

`get`
Ottieni una o più risorse da Trident.

`help`
Aiuto per qualsiasi comando.

`images`
Stampa una tabella delle immagini dei contenitori di cui Trident ha bisogno.

`import`
Importa una risorsa esistente in Trident.

`install`
Installa Trident.

`logs`
Stampa i registri da Trident.

`send`
Invia una risorsa da Trident.

`uninstall`
Disinstallare Trident.

`update`
Modificare una risorsa in Trident.

`update backend state`
Sospendere temporaneamente le operazioni di backend.

`upgrade`
Aggiorna una risorsa in Trident.

`version`
Stampa la versione di Trident.

Bandiere globali

-d, --debug
Output di debug.

-h, --help
Aiuto per `tridentctl` .

-k, --kubeconfig string
Specificare il `KUBECONFIG` percorso per eseguire i comandi localmente o da un cluster Kubernetes a un altro.

In alternativa, è possibile esportare il `KUBECONFIG` variabile per puntare a un cluster Kubernetes specifico e problema `tridentctl` comandi a quel cluster.

-n, --namespace string
Spazio dei nomi della distribuzione Trident .

-o, --output string
Formato di output. Uno tra json|yaml|name|wide|ps (predefinito).

-s, --server string
Indirizzo/porta dell'interfaccia REST Trident .

L'interfaccia REST Trident può essere configurata per ascoltare e servire solo su 127.0.0.1 (per IPv4) o [::1] (per IPv6).

Opzioni e flag dei comandi

creare

Utilizzare il `create` comando per aggiungere una risorsa a Trident.

```
tridentctl create [option]
```

Opzioni

`backend`: Aggiungi un backend a Trident.

eliminare

Utilizzare il `delete` comando per rimuovere una o più risorse da Trident.

```
tridentctl delete [option]
```

Opzioni

`backend`: Elimina uno o più backend di archiviazione da Trident.
`snapshot` : Elimina uno o più snapshot del volume da Trident.
`storageclass` : Elimina una o più classi di archiviazione da Trident.

`volume` : Elimina uno o più volumi di archiviazione da Trident.

Ottenere

Utilizzare il `get` comando per ottenere una o più risorse da Trident.

```
tridentctl get [option]
```

Opzioni

`backend`: Ottieni uno o più backend di archiviazione da Trident.
`snapshot` : Ottieni uno o più snapshot da Trident.
`storageclass` : Ottieni una o più classi di archiviazione da Trident.
`volume` : Ottieni uno o più volumi da Trident.

Bandiere

`-h`, `--help` : Aiuto per i volumi.
`--parentOfSubordinate string` : Limita la query al volume sorgente subordinato.
`--subordinateOf string` : Limita la query ai subordinati del volume.

immagini

Utilizzo `images` flag per stampare una tabella delle immagini del contenitore di cui Trident ha bisogno.

```
tridentctl images [flags]
```

Bandiere

`-h`, `--help` : Aiuto per le immagini.
`-v` , `--k8s-version string` : Versione semantica del cluster Kubernetes.

volume di importazione

Utilizzare il `import volume` comando per importare un volume esistente in Trident.

```
tridentctl import volume <backendName> <volumeName> [flags]
```

Alias

`volume`, `v`

Bandiere

`-f`, `--filename string` : Percorso al file PVC YAML o JSON.
`-h` , `--help` : Aiuto per il volume.
`--no-manage` : Crea solo PV/PVC. Non dare per scontato che la gestione del ciclo di vita sia incentrata sul volume.

installare

Utilizzare il `install` flag per installare Trident.

```
tridentctl install [flags]
```

Bandiere

`--autosupport-image string`: L'immagine del contenitore per Autosupport Telemetry (predefinita "netapp/trident autosupport:<current-version>").
`--autosupport-proxy string` : Indirizzo/porta di un proxy per l'invio di dati di telemetria di Autosupport.
`--enable-node-prep` : Tentativo di installare i pacchetti richiesti sui nodi.
`--generate-custom-yaml` : Genera file YAML senza installare nulla.
`-h` , `--help` : Aiuto per l'installazione.
`--http-request-timeout` : Sostituisci il timeout della richiesta HTTP per l'API REST del controller Trident (predefinito 1m30s).
`--image-registry string` : L'indirizzo/porta di un registro di immagini interno.
`--k8s-timeout duration` : Timeout per tutte le operazioni Kubernetes (predefinito 3m0s).
`--kubelet-dir string` : Posizione host dello stato interno di kubelet (predefinito "/var/lib/kubelet").
`--log-format string` : Formato di registrazione Trident (testo, json) (predefinito "testo").
`--node-prep` : consente a Trident di preparare i nodi del cluster Kubernetes per gestire i volumi utilizzando il protocollo di archiviazione dati specificato. **Attualmente, `iscsi` è l'unico valore supportato. A partire da OpenShift 4.19, la versione minima Trident supportata per questa funzionalità è 25.06.1.**
`--pv string` : Il nome del PV legacy utilizzato da Trident, assicura che non esista (predefinito "trident").
`--pvc string` : Il nome del PVC legacy utilizzato da Trident, assicura che questo non esista (predefinito "trident").
`--silence-autosupport` : Non inviare automaticamente i bundle di supporto automatico a NetApp (valore predefinito: true).
`--silent` : Disabilita la maggior parte degli output durante l'installazione.
`--trident-image string` : L'immagine Trident da installare.
`--k8s-api-qps` : Limite di query al secondo (QPS) per le richieste API di Kubernetes (predefinito 100; facoltativo).
`--use-custom-yaml` : Utilizzare tutti i file YAML esistenti nella directory di installazione.
`--use-ipv6` : Utilizza IPv6 per la comunicazione di Trident.

registri

Utilizzo `logs` flag per stampare i log da Trident.

```
tridentctl logs [flags]
```

Bandiere

`-a`, `--archive` : Crea un archivio di supporto con tutti i log, salvo diversa indicazione.
`-h` , `--help` : Aiuto per i registri.
`-l` , `--log string` : Registro Trident da visualizzare. Uno tra trident|auto|trident-operator|all (predefinito "auto").
`--node string` : Nome del nodo Kubernetes da cui raccogliere i log dei pod dei nodi.
`-p` , `--previous` : Ottieni i log per l'istanza del contenitore precedente, se esiste.
`--sidecars` : Prendi i registri per i contenitori del sidecar.

Inviare

Utilizzare il `send` comando per inviare una risorsa da Trident.

```
tridentctl send [option]
```

Opzioni

`autosupport`: Invia un archivio Autosupport a NetApp.

disinstallare

Utilizzo `uninstall` flag per disinstallare Trident.

```
tridentctl uninstall [flags]
```

Bandiere

`-h, --help`: Aiuto per la disinstallazione.
`--silent` : Disabilita la maggior parte degli output durante la disinstallazione.

aggiornamento

Utilizzare il `update` comando per modificare una risorsa in Trident.

```
tridentctl update [option]
```

Opzioni

`backend`: Aggiorna un backend in Trident.

aggiorna lo stato del backend

Utilizzare il `update backend state` comando per sospendere o riprendere le operazioni di backend.

```
tridentctl update backend state <backend-name> [flag]
```

Punti da considerare

- Se un backend viene creato utilizzando un TridentBackendConfig (tbc), il backend non può essere aggiornato utilizzando un `backend.json` file.
- Se il `userState` è stato impostato in un tbc, non può essere modificato utilizzando `tridentctl update backend state <backend-name> --user-state suspended/normal` comando.
- Per riacquistare la capacità di impostare il `userState` tramite tridentctl dopo che è stato impostato tramite tbc, il `userState` il campo deve essere rimosso dal tbc. Questo può essere fatto utilizzando il `kubectl edit tbc` comando. Dopo il `userState` campo viene rimosso, è possibile utilizzare il `tridentctl update backend state` comando per cambiare il `userState` di un backend.
- Utilizzare il `tridentctl update backend state` per cambiare il `userState` . Puoi anche aggiornare il `userState` usando `TridentBackendConfig` O `backend.json` file; ciò innesca una reinizializzazione completa del backend e può richiedere molto tempo.

  Bandiere

  `-h, --help` : Aiuto per lo stato del backend.
  `--user-state` : Impostato su `suspended` per mettere in pausa le operazioni di backend. Impostato su `normal` per riprendere le operazioni di backend. Quando impostato su `suspended` :

- `AddVolume`E `Import Volume` sono in pausa.
- `CloneVolume`, `ResizeVolume` , `PublishVolume` , `UnPublishVolume` , `CreateSnapshot` , `GetSnapshot` , `RestoreSnapshot` , `DeleteSnapshot` , `RemoveVolume` , `GetVolumeExternal` ,

`ReconcileNodeAccess` rimangono disponibili.

Puoi anche aggiornare lo stato del backend utilizzando `userState` campo nel file di configurazione del backend `TridentBackendConfig` O `backend.json` . Per maggiori informazioni, fare riferimento a"Opzioni per la gestione dei backend" E"Eseguire la gestione del backend con kubectl" .

Esempio:

JSON

Segui questi passaggi per aggiornare il `userState` utilizzando il `backend.json` file:

1. Modifica il `backend.json` file da includere `userState` campo con il suo valore impostato su "sospeso".
2. Aggiorna il backend utilizzando `tridentctl update backend` comando e il percorso per l'aggiornamento `backend.json` file.

   Esempio: `tridentctl update backend -f /<path to backend JSON file>/backend.json -n trident`

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "<redacted>",
  "svm": "nas-svm",
  "backendName": "customBackend",
  "username": "<redacted>",
  "password": "<redacted>",
  "userState": "suspended"
}
```

YAML

È possibile modificare il tbc dopo averlo applicato utilizzando `kubectl edit <tbc-name> -n <namespace>` comando. L'esempio seguente aggiorna lo stato del backend per sospendere utilizzando `userState: suspended` opzione:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  backendName: customBackend
  storageDriverName: ontap-nas
  managementLIF: <redacted>
  svm: nas-svm
  userState: suspended
  credentials:
    name: backend-tbc-ontap-nas-secret
```

versione

Utilizzo `version` bandiere per stampare la versione di `tridentctl` e il servizio Trident in funzione.

```
tridentctl version [flags]
```

Bandiere

`--client`: Solo versione client (non è richiesto alcun server).
`-h, --help` : Aiuto per la versione.

Supporto plugin

Tridentctl supporta plugin simili a kubectl. Tridentctl rileva un plugin se il nome del file binario del plugin segue lo schema "tridentctl-<plugin>" e il binario si trova in una cartella elencata nella variabile di ambiente PATH. Tutti i plugin rilevati sono elencati nella sezione plugin della guida di tridentctl. Facoltativamente, puoi anche limitare la ricerca specificando una cartella di plugin nella variabile d'ambiente TRIDENTCTL_PLUGIN_PATH (esempio: `TRIDENTCTL_PLUGIN_PATH=~/tridentctl-plugins/`). Se si utilizza la variabile, tridenctl effettua la ricerca solo nella cartella specificata.

Monitor Trident

Trident fornisce un set di endpoint di metriche Prometheus che puoi utilizzare per monitorare le prestazioni Trident .

Panoramica

Le metriche fornite da Trident consentono di fare quanto segue:

- Tieni d'occhio lo stato di salute e la configurazione di Trident. È possibile verificare il successo delle operazioni e se è possibile comunicare con i backend come previsto.
- Esaminare le informazioni sull'utilizzo del backend e comprendere quanti volumi sono forniti su un backend, la quantità di spazio consumata e così via.
- Mantenere una mappatura della quantità di volumi forniti sui backend disponibili.
- Prestazioni in pista. Puoi dare un'occhiata a quanto tempo impiega Trident per comunicare con i backend ed eseguire le operazioni.

> Per impostazione predefinita, le metriche di Trident sono esposte sulla porta di destinazione `8001` al `/metrics` punto finale. Queste metriche sono **abilitate per impostazione predefinita** quando Trident è installato.

Cosa ti servirà

- Un cluster Kubernetes con Trident installato.
- Un esempio di Prometeo. Questo può essere un "distribuzione Prometheus containerizzata" oppure puoi scegliere di eseguire Prometheus come "applicazione nativa" .

Fase 1: definire un obiettivo Prometheus

Dovresti definire un target Prometheus per raccogliere le metriche e ottenere informazioni sui backend gestiti Trident , sui volumi che crea e così via. Questo "blog" spiega come utilizzare Prometheus e Grafana con Trident per recuperare le metriche. Il blog spiega come eseguire Prometheus come operatore nel cluster

Kubernetes e come creare un ServiceMonitor per ottenere le metriche Trident .

Passaggio 2: creare un Prometheus ServiceMonitor

Per utilizzare le metriche Trident , è necessario creare un Prometheus ServiceMonitor che monitora il `trident-csi` servizio e ascolta sul `metrics` porta. Un esempio di ServiceMonitor si presenta così:

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: trident-sm
  namespace: monitoring
  labels:
    release: prom-operator
spec:
  jobLabel: trident
  selector:
    matchLabels:
      app: controller.csi.trident.netapp.io
  namespaceSelector:
    matchNames:
      - trident
  endpoints:
    - port: metrics
      interval: 15s
```

Questa definizione di ServiceMonitor recupera le metriche restituite da `trident-csi` servizio e cerca specificamente il `metrics` endpoint del servizio. Di conseguenza, Prometheus è ora configurato per comprendere le metriche di Trident.

Oltre alle metriche disponibili direttamente da Trident, kubelet espone molti `kubelet_volume_*` metriche tramite il proprio endpoint metrico. Kubelet può fornire informazioni sui volumi collegati, sui pod e sulle altre operazioni interne che gestisce. Fare riferimento a "Qui" .

Passaggio 3: interrogare le metriche Trident con PromQL

PromQL è utile per creare espressioni che restituiscono serie temporali o dati tabulari.

Ecco alcune query PromQL che puoi utilizzare:

Ottieni informazioni sulla salute Trident

- **Percentuale di risposte HTTP 2XX da Trident**

```
(sum (trident_rest_ops_seconds_total_count{status_code=~"2.."} OR on()
vector(0)) / sum (trident_rest_ops_seconds_total_count)) * 100
```

- **Percentuale di risposte REST da Trident tramite codice di stato**

```
(sum (trident_rest_ops_seconds_total_count) by (status_code)  / scalar
(sum (trident_rest_ops_seconds_total_count))) * 100
```

- **Durata media in ms delle operazioni eseguite da Trident**

```
sum by (operation)
(trident_operation_duration_milliseconds_sum{success="true"}) / sum by
(operation)
(trident_operation_duration_milliseconds_count{success="true"})
```

Ottieni informazioni sull'utilizzo Trident

- **Dimensione media del volume**

```
trident_volume_allocated_bytes/trident_volume_count
```

- **Spazio totale del volume fornito da ciascun backend**

```
sum (trident_volume_allocated_bytes) by (backend_uuid)
```

Ottieni l'utilizzo del volume individuale

Questa opzione è abilitata solo se vengono raccolte anche le metriche kubelet.

- **Percentuale di spazio utilizzato per ogni volume**

```
kubelet_volume_stats_used_bytes / kubelet_volume_stats_capacity_bytes *
100
```

Scopri di più sulla telemetria di Trident AutoSupport

Per impostazione predefinita, Trident invia le metriche di Prometheus e le informazioni di base del backend a NetApp con cadenza giornaliera.

- Per impedire a Trident di inviare metriche Prometheus e informazioni di base sul backend a NetApp, passare il `--silence-autosupport` bandiera durante l'installazione Trident .
- Trident può anche inviare i log dei container al supporto NetApp su richiesta tramite `tridentctl send autosupport` . Sarà necessario attivare Trident per caricare i suoi registri. Prima di inviare i log, dovresti accettare i termini e le condizioni di NetApphttps://www.netapp.com/company/legal/privacy-policy/["politica sulla riservatezza"^] .

- Se non diversamente specificato, Trident recupera i log delle ultime 24 ore.
- È possibile specificare l'intervallo di tempo di conservazione del registro con `--since` bandiera. Per esempio: `tridentctl send autosupport --since=1h` . Queste informazioni vengono raccolte e inviate tramite un `trident-autosupport` contenitore installato accanto a Trident. È possibile ottenere l'immagine del contenitore su "Trident AutoSupport" .
- Trident AutoSupport non raccoglie né trasmette informazioni di identificazione personale (PII) o informazioni personali. Viene fornito con un "Contratto di licenza con l'utente finale" ciò non è applicabile all'immagine del contenitore Trident stesso. Puoi scoprire di più sull'impegno di NetApp per la sicurezza e l'affidabilità dei dati "Qui" .

Un esempio di payload inviato da Trident si presenta così:

```
---
items:
  - backendUUID: ff3852e1-18a5-4df4-b2d3-f59f829627ed
    protocol: file
    config:
      version: 1
      storageDriverName: ontap-nas
      debug: false
      debugTraceFlags: null
      disableDelete: false
      serialNumbers:
        - nwkvzfanek_SN
      limitVolumeSize: ""
    state: online
    online: true
```

- I messaggi AutoSupport vengono inviati all'endpoint AutoSupport di NetApp. Se si utilizza un registro privato per archiviare le immagini dei contenitori, è possibile utilizzare `--image-registry` bandiera.
- È anche possibile configurare gli URL proxy generando i file YAML di installazione. Questo può essere fatto utilizzando `tridentctl install --generate-custom-yaml` per creare i file YAML e aggiungere il `--proxy-url` argomento per la `trident-autosupport` contenitore in `trident-deployment.yaml` .

Disabilita le metriche Trident

Per **disabilitare** la segnalazione delle metriche, dovresti generare YAML personalizzati (utilizzando `--generate-custom-yaml` flag) e modificarli per rimuovere il `--metrics` flag dall'essere invocato per il `trident-main` contenitore.

Disinstallare Trident

Per disinstallare Trident dovresti usare lo stesso metodo che hai usato per installare Trident.

Informazioni su questo compito

- Se hai bisogno di una correzione per i bug osservati dopo un aggiornamento, problemi di dipendenza o un aggiornamento non riuscito o incompleto, dovresti disinstallare Trident e reinstallare la versione precedente utilizzando le istruzioni specifiche per quella"versione" . Questo è l'unico metodo consigliato per effettuare il *downgrade* a una versione precedente.
- Per facilitare l'aggiornamento e la reinstallazione, la disinstallazione Trident non rimuove i CRD o gli oggetti correlati creati da Trident. Se è necessario rimuovere completamente Trident e tutti i suoi dati, fare riferimento a"Rimuovere completamente Trident e CRD" .

Prima di iniziare

Se si desidera dismettere i cluster Kubernetes, è necessario eliminare tutte le applicazioni che utilizzano volumi creati da Trident prima della disinstallazione. Ciò garantisce che i PVC non vengano pubblicati sui nodi Kubernetes prima di essere eliminati.

Determinare il metodo di installazione originale

Per disinstallare Trident dovresti usare lo stesso metodo che hai usato per installarlo. Prima di disinstallare, verifica quale versione hai utilizzato per installare originariamente Trident.

1. Utilizzo `kubectl get pods -n trident` per esaminare i baccelli.
   - Se non è presente alcun pod operatore, Trident è stato installato utilizzando `tridentctl` .
   - Se è presente un pod operatore, Trident è stato installato utilizzando l'operatore Trident manualmente o tramite Helm.
2. Se è presente un pod operatore, utilizzare `kubectl describe tproc trident` per determinare se Trident è stato installato tramite Helm.
   - Se è presente un'etichetta Helm, Trident è stato installato tramite Helm.
   - Se non è presente alcuna etichetta Helm, Trident è stato installato manualmente utilizzando l'operatore Trident .

Disinstallare un'installazione dell'operatore Trident

È possibile disinstallare manualmente un'installazione dell'operatore Trident oppure tramite Helm.

Disinstallare l'installazione manuale

Se hai installato Trident tramite l'operatore, puoi disinstallarlo eseguendo una delle seguenti operazioni:

1. **Modificare `TridentOrchestrator` CR e imposta il flag di disinstallazione**:

   ```
   kubectl patch torc <trident-orchestrator-name> --type=merge -p
   '{"spec":{"uninstall":true}}'
   ```

   Quando il `uninstall` la bandiera è impostata su `true` , l'operatore Trident disinstalla Trident, ma non rimuove TridentOrchestrator stesso. Se vuoi installare nuovamente Trident, dovresti pulire TridentOrchestrator e crearne uno nuovo.

2. **Eliminare `TridentOrchestrator`** : Rimuovendo il `TridentOrchestrator` CR utilizzato per distribuire Trident, si chiede all'operatore di disinstallare Trident. L'operatore elabora la rimozione di `TridentOrchestrator` e procede alla rimozione della distribuzione e del daemonset Trident ,

eliminando i pod Trident creati come parte dell'installazione.

```
kubectl delete -f deploy/<bundle.yaml> -n <namespace>
```

Disinstallare l'installazione di Helm

Se hai installato Trident tramite Helm, puoi disinstallarlo tramite `helm uninstall`.

```
#List the Helm release corresponding to the Trident install.
helm ls -n trident
NAME          NAMESPACE       REVISION        UPDATED
STATUS          CHART                           APP VERSION
trident       trident         1               2021-04-20
00:26:42.417764794 +0000 UTC deployed        trident-operator-21.07.1
21.07.1

#Uninstall Helm release to remove Trident
helm uninstall trident -n trident
release "trident" uninstalled
```

Disinstallare un `tridentctl` installazione

Utilizzare il `uninstall` comando in `tridentctl` per rimuovere tutte le risorse associate a Trident, ad eccezione dei CRD e degli oggetti correlati:

```
./tridentctl uninstall -n <namespace>
```

Informazioni sul copyright

Informazioni sul marchio commerciale