Gestisci e monitora Astra Trident

Astra Trident

NetApp
July 12, 2024

Sommario

Gestisci e monitora Astra Trident

Aggiorna Astra Trident

Aggiorna Astra Trident

A partire dalla release 24,02, Astra Trident segue una cadenza di quattro mesi, fornendo tre release principali ogni anno solare. Ogni nuova release si basa sulle release precedenti e offre nuove funzionalità, miglioramenti delle prestazioni, correzioni di bug e miglioramenti. Ti consigliamo di effettuare l'upgrade almeno una volta all'anno per sfruttare le nuove funzionalità di Astra Trident.

Considerazioni prima dell'aggiornamento

Quando si esegue l'aggiornamento all'ultima release di Astra Trident, considerare quanto segue:

- Dovrebbe essere installata una sola istanza di Astra Trident in tutti gli spazi dei nomi di un determinato cluster Kubernetes.
- Astra Trident 23,07 e versioni successive richiede snapshot di volume v1 e non supporta più snapshot alfa o beta.
- Se hai creato Cloud Volumes Service per Google Cloud nel "Tipo di servizio CVS", è necessario aggiornare la configurazione del backend per utilizzare `standardsw` oppure `zoneredundantstandardsw` Livello di servizio durante l'upgrade da Astra Trident 23,01. Impossibile aggiornare `serviceLevel` nel backend potrebbe causare un errore nei volumi. Fare riferimento a. "Esempi di tipo di servizio CVS" per ulteriori informazioni.
- Durante l'aggiornamento, è importante fornire `parameter.fsType` poll `StorageClasses` Utilizzato da Astra Trident. È possibile eliminare e ricreare `StorageClasses` senza interrompere i volumi preesistenti.
  - Si tratta di un requisito ** per l'applicazione "contesti di sicurezza" Per volumi SAN.
  - La directory sample input contiene esempi, come `storage-class-basic.yaml.templ` e `storage-class-bronze-default.yaml`.
  - Per ulteriori informazioni, fare riferimento a. "Problemi noti".

Fase 1: Selezionare una versione

Le versioni di Astra Trident seguono una data-based `YY.MM` Convenzione di naming, dove "YY" è l'ultima cifra dell'anno e "MM" è il mese. I rilasci di punti seguono un `YY.MM.X` convention, dove "X" è il livello di patch. Selezionare la versione a cui eseguire l'aggiornamento in base alla versione da cui si sta eseguendo l'aggiornamento.

- È possibile eseguire un aggiornamento diretto a qualsiasi release di destinazione che si trova all'interno di una finestra di quattro release della versione installata. Ad esempio, è possibile aggiornare direttamente da 23,04 (o qualsiasi versione a 23,04 punti) a 24,06.
- Se si sta eseguendo l'aggiornamento da una release al di fuori della finestra a quattro release, eseguire un aggiornamento in più fasi. Utilizzare le istruzioni di aggiornamento per il "versione precedente" quale si sta eseguendo l'aggiornamento per passare alla versione più recente adatta alla finestra a quattro release. Ad esempio, se si utilizza 22,01 e si desidera eseguire l'aggiornamento a 24,06:
  a. Primo aggiornamento da 22,07 a 23,04.

b. Quindi, eseguire l'aggiornamento da 23,04 a 24,06.

> Quando si esegue l'aggiornamento utilizzando 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 21.01.0 contiene un problema noto che è stato risolto nel 21.01.1. Per ulteriori informazioni, fare riferimento alla "Dettagli del problema su GitHub".

Fase 2: Determinare il metodo di installazione originale

Per determinare la versione utilizzata per installare Astra Trident:

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

Fase 3: Selezionare un metodo di aggiornamento

In genere, è necessario eseguire l'aggiornamento utilizzando lo stesso metodo utilizzato per l'installazione iniziale, tuttavia è possibile "passare da un metodo di installazione all'altro". Ci sono due opzioni per aggiornare Astra Trident.

- "Eseguire l'aggiornamento utilizzando l'operatore Trident"

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

*

Eseguire l'upgrade con l'operatore

Comprendere il flusso di lavoro di aggiornamento dell'operatore

Prima di utilizzare l'operatore Trident per aggiornare Astra Trident, devi comprendere i processi in background che si verificano durante l'upgrade. Sono incluse le modifiche al controller Trident, al pod dei controller e ai pod dei nodi e al daemonSet dei nodi che consentono l'esecuzione degli aggiornamenti.

Gestione dell'aggiornamento dell'operatore Trident

Uno dei tanti "Vantaggi dell'utilizzo dell'operatore Trident" Per installare e aggiornare Astra Trident è la gestione automatica degli oggetti Astra Trident e Kubernetes senza interrompere i volumi montati. In questo modo, Astra Trident può supportare gli upgrade senza downtime o. "*rolling updates*". In particolare, l'operatore Trident comunica con il cluster Kubernetes per:

- Eliminare e ricreare l'implementazione del controller Trident e il daemonSet del nodo.
- Sostituisci il Controller Pod Trident e i pod di nodi Trident con nuove versioni.
  - Se un nodo non viene aggiornato, non impedisce l'aggiornamento dei nodi rimanenti.
  - Solo i nodi con un pod nodo Trident in esecuzione possono montare volumi.

Per ulteriori informazioni sull'architettura Astra Trident nel cluster Kubernetes, fare riferimento a. "Architettura Astra 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 attualmente installata di Astra Trident (versione *n*).
   b. Aggiorna tutti gli oggetti Kubernetes, inclusi CRD, RBAC e Trident SVC.
   c. Elimina l'implementazione del controller Trident per la versione *n*.
   d. Crea l'implementazione del controller Trident per la versione *n+1*.
2. **Kubernetes** crea il Pod controller Trident per *n+1*.
3. L'operatore **Trident**:
   a. Elimina il daemonSet del nodo Trident per *n*. L'operatore non attende la terminazione del nodo Pod.
   b. Crea il nodo Trident Daemonset per *n+1*.
4. **Kubernetes** crea pod di nodi Trident sui nodi che non eseguono il pod di nodi Trident *n*. In questo modo, si garantisce che non ci sia mai più di un Pod nodi Trident, di qualsiasi versione, su un nodo.

Aggiornare un'installazione Astra Trident usando l'operatore Trident o Helm

È possibile eseguire l'aggiornamento di Astra Trident utilizzando l'operatore Trident sia manualmente che tramite Helm. È possibile eseguire l'aggiornamento da un'installazione dell'operatore Trident a un'altra installazione dell'operatore Trident o da un `tridentctl` Installazione su una versione dell'operatore Trident. Revisione "Selezionare un metodo di aggiornamento" Prima di aggiornare un'installazione dell'operatore Trident.

Aggiornare un'installazione manuale

È possibile eseguire l'aggiornamento da un'installazione dell'operatore Trident definita dall'ambito del cluster a un'altra installazione dell'operatore Trident definita dal cluster. Tutte le versioni di Astra Trident 21.01 e successive utilizzano un operatore con ambito cluster.

Per l'aggiornamento da Astra Trident che è stato installato usando l'operatore con ambito namespace (versioni da 20,07 a 20,10), usa le istruzioni di upgrade per "versione installata" Di Astra Trident.

A proposito di questa attività

Trident fornisce un file bundle da utilizzare per installare l'operatore e creare oggetti associati per la 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 versione successiva, utilizzare "bundle_post_1_25.yaml".

Prima di iniziare

Assicurarsi di utilizzare un cluster Kubernetes in esecuzione "Una versione di Kubernetes supportata".

Fasi

1. Verificare la versione di Astra Trident:

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

2. Eliminare l'operatore Trident utilizzato per installare l'istanza corrente di Astra Trident. Ad esempio, se si sta eseguendo l'aggiornamento da 23,07, eseguire il seguente comando:

```
kubectl delete -f 23.07.0/trident-installer/deploy/<bundle.yaml> -n
trident
```

3. Se l'installazione iniziale è stata personalizzata utilizzando `TridentOrchestrator` è possibile modificare `TridentOrchestrator` oggetto per modificare i parametri di installazione. Ciò potrebbe includere le modifiche apportate per specificare i registri di immagini Trident e CSI mirrorati per la modalità offline, abilitare i registri di debug o specificare i segreti di pull delle immagini.
4. Installa Astra Trident usando il file YAML del bundle corretto per il tuo ambiente, dove *<bundle.yaml>* si trova `bundle_pre_1_25.yaml` o `bundle_post_1_25.yaml` si basa sulla tua versione di Kubernetes. Ad esempio, se stai installando Astra Trident 24,06, esegui il seguente comando:

```
kubectl create -f 24.06.0/trident-installer/deploy/<bundle.yaml> -n
trident
```

Aggiornare un'installazione Helm

È possibile aggiornare un'installazione di Astra Trident Helm.

> Quando si aggiorna un cluster Kubernetes dalla versione 1.24 alla 1.25 o successiva su cui è installato Astra Trident, è necessario aggiornare values.yaml per impostarlo `excludePodSecurityPolicy` a. `true` oppure aggiungi `--set excludePodSecurityPolicy=true` al `helm upgrade` prima di aggiornare il cluster.

Fasi

1. Se si "Installato Astra Trident utilizzando Helm"utilizza , è possibile utilizzare `helm upgrade trident netapp-trident/trident-operator --version 100.2406.0` per eseguire l'aggiornamento in un solo passaggio. Se non è stato aggiunto il repo Helm o non è possibile utilizzarlo per l'aggiornamento:
   a. Scarica la versione più recente di Astra Trident da "La sezione *Assets* su GitHub".
   b. Utilizzare il `helm upgrade` comando dove riflette la versione a cui `trident-operator-24.06.0.tgz` si desidera eseguire l'aggiornamento.

```
helm upgrade <name> trident-operator-24.06.0.tgz
```

> Se si impostano opzioni personalizzate durante l'installazione iniziale (ad esempio, se si specificano registri privati con mirroring per le immagini Trident e CSI), aggiungere il `helm upgrade` utilizzare `--set` per assicurarsi che tali opzioni siano incluse nel comando upgrade, altrimenti i valori torneranno ai valori predefiniti.

2. Eseguire `helm list` per verificare che la versione del grafico e dell'applicazione sia stata aggiornata. Eseguire `tridentctl logs` per esaminare eventuali messaggi di debug.

Aggiornamento da a. `tridentctl` Installazione all'operatore Trident

È possibile eseguire 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, vedere "Passaggio da un metodo di installazione all'altro".

Fasi

1. Scarica l'ultima release di Astra Trident.

```
# Download the release required [24.060.0]
mkdir 24.06.0
cd 24.06.0
wget
https://github.com/NetApp/trident/releases/download/v24.06.0/trident-
installer-24.06.0.tar.gz
tar -xf trident-installer-24.06.0.tar.gz
cd trident-installer
```

2. Creare il `tridentorchestrator` CRD dal manifesto.

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

3. Implementare 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. Creare un `TridentOrchestrator` CR per l'installazione di Astra 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. Confermare che Trident è stato aggiornato alla versione prevista.

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

Message:                Trident installed
Namespace:              trident
Status:                 Installed
Version:                v24.06.0
```

Upgrade con tridentctl

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

A proposito di questa attività

La disinstallazione e la reinstallazione di Astra Trident funge da aggiornamento. Quando si disinstalla Trident, i PVC (Persistent Volume Claim) e PV (Persistent Volume) utilizzati dall'implementazione di Astra Trident non vengono cancellati. I PVS già forniti resteranno disponibili mentre Astra Trident è offline e Astra Trident effettuerà il provisioning dei volumi per i PVC creati nel frattempo una volta tornati online.

Prima di iniziare

Revisione "Selezionare un metodo di aggiornamento" prima di eseguire l'aggiornamento con `tridentctl`.

Fasi

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

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

2. Reinstallare Astra Trident. Fare riferimento a. "Installare Astra Trident usando tridentctl".

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

Gestisci Astra Trident usando tridentctl

Il "Pacchetto di installazione Trident" include il `tridentctl` Command-line utility per fornire un semplice accesso ad Astra Trident. Gli utenti Kubernetes con privilegi sufficienti possono usarlo per installare Astra Trident o gestire il namespace che contiene il pod Astra Trident.

Comandi e flag globali

Puoi correre `tridentctl help` per ottenere un elenco di 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]
```

Astra Trident `tridentctl` utility supporta i seguenti comandi e flag globali.

Comandi

`create`
Aggiungi una risorsa a Astra Trident.

`delete`
Rimozione di una o più risorse da Astra Trident.

`get`
Ottieni una o più risorse da Astra Trident.

`help`
Aiuto su qualsiasi comando.

`images`
Stampare una tabella delle immagini container di cui Astra Trident ha bisogno.

`import`
Importa una risorsa esistente in Astra Trident.

`install`
Installa Astra Trident.

`logs`
Stampare i registri da Astra Trident.

`send`
Invia una risorsa da Astra Trident.

`uninstall`
Disinstallare Astra Trident.

`update`
Modifica una risorsa in Astra Trident.

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

`upgrade`
Aggiorna una risorsa in Astra Trident.

`version`
Stampa la versione di Astra Trident.

Flag globali

-d, --debug
Output di debug.

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

-k, --kubeconfig string
Specificare `KUBECONFIG` Percorso per eseguire comandi in locale o da un cluster Kubernetes a un altro.

> In alternativa, è possibile esportare `KUBECONFIG` Variabile che indica un problema e un cluster Kubernetes specifici `tridentctl` comandi a quel cluster.

-n, --namespace string
Namespace dell'implementazione di Astra Trident.

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

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

> L'interfaccia REST di Trident può essere configurata per l'ascolto e la distribuzione solo su 127.0.0.1 (per IPv4) o [::1] (per IPv6).

Opzioni di comando e flag

creare

Utilizzare `create` Comando per aggiungere una risorsa ad Astra Trident.

```
tridentctl create [option]
```

Opzioni

`backend`: Aggiungi un backend ad Astra Trident.

eliminare

Utilizzare `delete` Comando per rimuovere una o più risorse da Astra Trident.

```
tridentctl delete [option]
```

Opzioni

`backend`: Eliminare uno o più backend di storage da Astra Trident.
`snapshot`: Consente di eliminare una o più snapshot di volumi da Astra Trident.
`storageclass`: Eliminare una o più classi di storage da Astra Trident.

`volume`: Eliminare uno o più volumi di storage da Astra Trident.

ottieni

Utilizzare `get` Comando per ottenere una o più risorse da Astra Trident.

```
tridentctl get [option]
```

Opzioni

`backend`: Ottieni uno o più backend di storage da Astra Trident.
`snapshot`: Ottenere una o più istantanee da Astra Trident.
`storageclass`: Ottieni una o più classi di storage da Astra Trident.
`volume`: Procurarsi uno o più volumi da Astra Trident.

Allarmi

`-h`, `--help`: Guida per i volumi.
`--parentOfSubordinate string`: Limita query al volume di origine subordinato.
`--subordinateOf string`: Limita la query alle subordinate del volume.

immagini

Utilizzare `images` Flag per stampare una tabella delle immagini container di cui Astra Trident ha bisogno.

```
tridentctl images [flags]
```

Allarmi

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

importa volume

Utilizzare `import volume` Comando per importare un volume esistente in Astra Trident.

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

Alias

`volume`, `v`

Allarmi

`-f`, `--filename string`: Percorso al file PVC YAML o JSON.
`-h`, `--help`: Guida per il volume.
`--no-manage`: Crea solo PV/PVC. Non presupporre la gestione del ciclo di vita dei volumi.

installare

Utilizzare `install` Flag per installare Astra Trident.

```
tridentctl install [flags]
```

Allarmi

`--autosupport-image string`: L'immagine contenitore per la telemetria AutoSupport (predefinita "netapp/trident autosupport:<current-version>").
`--autosupport-proxy string`Indirizzo/porta di un proxy per l'invio di telemetria AutoSupport.
`--enable-node-prep`: Tentare di installare i pacchetti richiesti sui nodi.
`--generate-custom-yaml`: Generare file YAML senza installare nulla.
`-h`, `--help`: Guida all'installazione.
`--http-request-timeout`: Ignorare il timeout della richiesta HTTP per l'API REST del controller Trident (valore predefinito 1m30).
`--image-registry string`: L'indirizzo/porta di un registro di immagini interno.
`--k8s-timeout duration`: Il timeout per tutte le operazioni Kubernetes (valore predefinito: 3 m0s).
`--kubelet-dir string`: La posizione host dello stato interno di kubelet (default "/var/lib/kubelet").
`--log-format string`: Il formato di registrazione Astra Trident (text, json) (default "text").
`--pv string`: Il nome del PV legacy utilizzato da Astra Trident, garantisce che non esista (il "tridente" predefinito).
`--pvc string`: Il nome del PVC legacy utilizzato da Astra Trident, garantisce che non esista (il "tridente" predefinito).
`--silence-autosupport`: Non inviare pacchetti AutoSupport a NetApp automaticamente (valore predefinito vero).
`--silent`: Disattivare l'output MOST durante l'installazione.
`--trident-image string`: L'immagine Astra Trident da installare.
`--use-custom-yaml`: Utilizzare tutti i file YAML esistenti nella directory di installazione.
`--use-ipv6`: Utilizza IPv6 per la comunicazione di Astra Trident.

registri

Utilizzare `logs` Flag per stampare i log da Astra Trident.

```
tridentctl logs [flags]
```

Allarmi

`-a`, `--archive`: Creare un archivio di supporto con tutti i log, se non diversamente specificato.
`-h`, `--help`: Guida per i log.
`-l`, `--log string`: Registro Astra Trident da visualizzare. Uno tra trident|auto|trident-operator|all (impostazione predefinita "auto").
`--node string`: Il nome del nodo Kubernetes da cui raccogliere i log dei pod dei nodi.
`-p`, `--previous`: Ottenere i log per l'istanza di container precedente, se esistente.
`--sidecars`: Ottenere i log per i contenitori del sidecar.

invia

Utilizzare `send` Comando per inviare una risorsa da Astra Trident.

```
tridentctl send [option]
```

Opzioni

`autosupport`: Inviare un archivio AutoSupport a NetApp.

disinstallazione

Utilizzare `uninstall` Flag per disinstallare Astra Trident.

```
tridentctl uninstall [flags]
```

Allarmi

`-h, --help`: Guida per la disinstallazione.
`--silent`: Disattivare la maggior parte dell'output durante la disinstallazione.

aggiornamento

Utilizzare `update` Comando per modificare una risorsa in Astra Trident.

```
tridentctl update [option]
```

Opzioni

`backend`: Aggiornare un backend in Astra Trident.

aggiorna stato backend

Utilizzare `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), non è possibile aggiornare il backend utilizzando un `backend.json` file.
- Se il `userState` è stato impostato in un tbc, non può essere modificato utilizzando il `tridentctl update backend state <backend-name> --user-state suspended/normal` comando .
- Per recuperare la capacità di impostare il `userState` tridentctl via una volta che è stato impostato tramite tbc, il `userState` campo deve essere rimosso dal tbc. Questo può essere fatto usando il `kubectl edit tbc` comando. Una volta rimosso il `userState` campo, è possibile utilizzare il `tridentctl update backend state` comando per modificare il `userState` di un backend.
- Utilizzare il `tridentctl update backend state` per modificare il `userState`. È anche possibile aggiornare il `userState` file Using `TridentBackendConfig` o `backend.json` ; questo attiva una reinizializzazione completa del backend e può richiedere molto tempo.

  #### Allarmi

  `-h, --help`: Guida per lo stato backend.
  `--user-state`: Impostare su `suspended` per sospendere le operazioni di backend. Impostare su `normal` per riprendere le operazioni di backend. Quando è impostato su `suspended`:

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

È inoltre possibile aggiornare lo stato backend utilizzando il `userState` campo nel file di configurazione backend `TridentBackendConfig` o `backend.json`. Per ulteriori informazioni, fare riferimento a "Opzioni

per la gestione dei backend" e "Eseguire la gestione del back-end con kubectl".

Esempio:

JSON

Per aggiornare utilizzando il file, procedere come segue `userState` `backend.json` :

1. Modificare il `backend.json` file per includere il `userState` campo con il valore impostato su 'sospeso'.
2. Aggiornare il backend utilizzando il `tridentctl backend update` comando e il percorso del file aggiornato `backend.json` .

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

```
{
    "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 il `kubectl edit <tbc-name> -n <namespace>` comando . Nell'esempio riportato di seguito viene aggiornato lo stato backend per la sospensione mediante l' `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

Utilizzare `version` contrassegni per stampare la versione di `tridentctl` E il servizio Running Trident.

```
tridentctl version [flags]
```

Allarmi

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

Monitorare Astra Trident

Astra Trident offre un set di endpoint di metriche Prometheus che è possibile utilizzare per monitorare le performance di Astra Trident.

Panoramica

Le metriche fornite da Astra Trident ti consentono di:

- Tieni sotto controllo lo stato di salute e la configurazione di Astra Trident. È possibile esaminare il successo delle operazioni e se è in grado di comunicare con i back-end come previsto.
- Esaminare le informazioni sull'utilizzo del back-end e comprendere il numero di volumi sottoposti a provisioning su un back-end, la quantità di spazio consumato e così via.
- Mantenere una mappatura della quantità di volumi forniti sui backend disponibili.
- Tenere traccia delle performance. Puoi dare un'occhiata a quanto tempo ci vuole per Astra Trident per comunicare con i back-end ed eseguire le operazioni.

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

Di cosa hai bisogno

- Un cluster Kubernetes con Astra Trident installato.
- Un'istanza Prometheus. Questo può essere un "Implementazione di Prometheus in container" Oppure puoi scegliere di eseguire Prometheus come a. "applicazione nativa".

Fase 1: Definire un target Prometheus

Devi definire un target Prometheus per raccogliere le metriche e ottenere informazioni sui backend gestiti da Astra Trident, sui volumi creati e così via. Questo "blog" Spiega come utilizzare Prometheus e Grafana con Astra Trident per recuperare le metriche. Il blog spiega come eseguire Prometheus come operatore nel cluster Kubernetes e la creazione di un ServiceMonitor per ottenere le metriche Astra Trident.

Fase 2: Creazione di un ServiceMonitor Prometheus

Per utilizzare le metriche Trident, è necessario creare un ServiceMonitor Prometheus che controlli `trident-csi` e ascolta su `metrics` porta. Un esempio di ServiceMonitor è simile al seguente:

```
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` e in particolare cerca di `metrics` endpoint del servizio. Di conseguenza, Prometheus è ora configurato per comprendere le metriche di Astra Trident.

Oltre alle metriche disponibili direttamente da Astra Trident, Kubelet ne espone molte `kubelet_volume_*` metriche tramite il proprio endpoint di metriche. Kubelet può fornire informazioni sui volumi collegati, sui pod e sulle altre operazioni interne gestite. Fare riferimento a. "qui".

Fase 3: Eseguire una query sulle metriche di Trident con PromQL

PromQL è utile per la creazione di espressioni che restituiscono dati di serie temporali o tabulari.

Di seguito sono riportate alcune query PromQL che è possibile utilizzare:

Ottieni informazioni sulla salute di Trident

- **Percentuale di risposte HTTP 2XX da Astra 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 Astra 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 Astra 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 di Astra 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 di singoli volumi

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

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

```
kubelet_volume_stats_used_bytes / kubelet_volume_stats_capacity_bytes *
100
```

Scopri di più sulla telemetria Astra Trident AutoSupport

Per impostazione predefinita, Astra Trident invia le metriche Prometheus e le informazioni di back-end di base a NetApp ogni giorno.

- Per impedire ad Astra Trident di inviare a NetApp le metriche Prometheus e le informazioni di back-end di base, passare il `--silence-autosupport` Segnalazione durante l'installazione di Astra Trident.
- Astra Trident può anche inviare i log dei container al NetApp Support on-demand tramite `tridentctl send autosupport`. Devi attivare Astra Trident per caricare i registri. Prima di inviare i log, è necessario accettare i file NetApphttps://www.netapp.com/company/legal/privacy-policy/["direttiva sulla privacy"^].
- Se non specificato, Astra Trident recupera i registri delle ultime 24 ore.
- È possibile specificare il periodo di conservazione dei log con `--since` allarme. Ad esempio: `tridentctl send autosupport --since=1h`. Queste informazioni vengono raccolte e inviate tramite un `trident-autosupport` Container installato insieme ad Astra Trident. È possibile ottenere l'immagine del contenitore in "Trident AutoSupport".
- Trident AutoSupport non raccoglie né trasmette dati personali o di identificazione personale (PII). Viene fornito con un "EULA" Non applicabile all'immagine del contenitore Trident. Scopri di più sull'impegno di

NetApp per la sicurezza e la fiducia dei dati "qui".

Un payload di esempio inviato da Astra Trident è simile al seguente:

```
---
items:
- backendUUID: ff3852e1-18a5-4df4-b2d3-f59f829627ed
  protocol: file
  config:
    version: 1
    storageDriverName: ontap-nas
    debug: false
    debugTraceFlags:
    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 memorizzare le immagini container, è possibile utilizzare `--image-registry` allarme.
- È inoltre possibile configurare gli URL proxy generando i file YAML di installazione. Per eseguire questa operazione, utilizzare `tridentctl install --generate-custom-yaml` Per creare i file YAML e aggiungere `--proxy-url` argomento per `trident-autosupport` container in `trident-deployment.yaml`.

Disattiva le metriche di Astra Trident

Per **disattivare** il report delle metriche, è necessario generare YAML personalizzati (utilizzando il `--generate-custom-yaml` e modificarli per rimuovere `--metrics` il contrassegno di non essere richiamato per `trident-main`container.

Disinstallare Astra Trident

Devi utilizzare lo stesso metodo per disinstallare Astra Trident che hai utilizzato per installare Astra Trident.

A proposito di questa attività

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

Prima di iniziare

Se stai decommissionando i cluster Kubernetes, devi eliminare tutte le applicazioni che utilizzano volumi creati da Astra Trident prima della disinstallazione. In questo modo, si garantisce che i PVC non siano pubblicati sui nodi Kubernetes prima di essere eliminati.

Determinare il metodo di installazione originale

Utilizzare lo stesso metodo per disinstallare Astra Trident che è stato utilizzato per installarlo. Prima di disinstallare, verificare quale versione è stata utilizzata per installare originariamente Astra Trident.

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

Disinstallare un'installazione dell'operatore Trident

È possibile disinstallare manualmente un'installazione dell'operatore tridente o utilizzando Helm.

Disinstallare l'installazione manuale

Se Astra Trident è stato installato utilizzando l'operatore, è possibile disinstallarlo effettuando una delle seguenti operazioni:

1. **Modifica `TridentOrchestrator` CR e impostare il flag di disinstallazione**:

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

   Quando il `uninstall` flag è impostato su `true`, L'operatore Trident disinstalla Trident, ma non rimuove il TridentOrchestrator stesso. Se si desidera installare di nuovo Trident, è necessario ripulire TridentOrchestrator e crearne uno nuovo.

2. **Elimina `TridentOrchestrator`**: Rimuovendo `TridentOrchestrator` CR utilizzato per implementare Astra Trident, si richiede all'operatore di disinstallare Trident. L'operatore elabora la rimozione di `TridentOrchestrator` E procede alla rimozione dell'implementazione e del demonset di Astra Trident, eliminando i pod Trident creati come parte dell'installazione.

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

Disinstallare l'installazione di Helm

Se Astra Trident è stato installato utilizzando Helm, è possibile disinstallarlo utilizzando `helm uninstall`.

```
#List the Helm release corresponding to the Astra 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 `uninstall` ingresso comando `tridentctl` Per rimuovere tutte le risorse associate a Astra Trident, ad eccezione dei CRD e degli oggetti correlati:

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

Informazioni sul copyright

Informazioni sul marchio commerciale