Buone pratiche e raccomandazioni

Trident

NetApp
April 08, 2026

Sommario

Buone pratiche e raccomandazioni

Distribuzione

Utilizza le raccomandazioni elencate qui quando distribuisci Trident.

Distribuisci in uno spazio dei nomi dedicato

"Spazi dei nomi" forniscono separazione amministrativa tra diverse applicazioni e rappresentano un ostacolo alla condivisione delle risorse. Ad esempio, un PVC di uno spazio dei nomi non può essere utilizzato da un altro. Trident fornisce risorse PV a tutti gli spazi dei nomi nel cluster Kubernetes e di conseguenza sfrutta un account di servizio con privilegi elevati.

Inoltre, l'accesso al pod Trident potrebbe consentire a un utente di accedere alle credenziali del sistema storage e ad altre informazioni sensibili. È importante assicurarsi che gli utenti dell'applicazione e le applicazioni di gestione non abbiano la possibilità di accedere alle definizioni degli oggetti Trident o ai pod stessi.

Utilizza le quote e i limiti di intervallo per controllare il consumo di storage

Kubernetes offre due funzionalità che, se combinate, forniscono un potente meccanismo per limitare il consumo di risorse da parte delle applicazioni. Il "meccanismo di storage quota" consente all'amministratore di implementare limiti di consumo di capacità e numero di oggetti, sia globali che specifici per classe di storage, su base per-namespace. Inoltre, l'utilizzo di un "limite di range" garantisce che le richieste PVC rientrino sia in un valore minimo che in un valore massimo prima che la richiesta venga inoltrata al provisioner.

Questi valori sono definiti per ogni namespace, il che significa che per ogni namespace devono essere definiti valori in linea con i relativi requisiti di risorse. Vedere qui per informazioni su "come sfruttare le quote".

Configurazione dello storage

Ogni piattaforma di storage nel portafoglio NetApp ha funzionalità uniche che avvantaggiano le applicazioni, containerizzate o meno.

Panoramica della piattaforma

Trident funziona con ONTAP ed Element. Non esiste una piattaforma più adatta a tutte le applicazioni e gli scenari rispetto a un'altra, tuttavia, nella scelta della piattaforma è necessario tenere conto delle esigenze dell'applicazione e del team che gestisce il dispositivo.

È consigliabile seguire le best practice di base per il sistema operativo host con il protocollo che si sta utilizzando. Facoltativamente, si potrebbe valutare l'integrazione delle best practice applicative, ove disponibili, con le impostazioni di backend, storage class e PVC per ottimizzare lo storage per applicazioni specifiche.

ONTAP e Cloud Volumes ONTAP best practices

Scopri le best practice per la configurazione di ONTAP e Cloud Volumes ONTAP per Trident.

Le seguenti raccomandazioni sono linee guida per la configurazione di ONTAP per carichi di lavoro containerizzati, che consumano volumi forniti dinamicamente da Trident. Ciascuna dovrebbe essere considerata e valutata per l'adeguatezza al proprio ambiente.

Utilizzare SVM dedicati a Trident

Le Storage Virtual Machines (SVM) forniscono isolamento e separazione amministrativa tra i tenant su un sistema ONTAP. Dedicare una SVM alle applicazioni consente la delega dei privilegi e consente di applicare le best practice per limitare il consumo di risorse.

Sono disponibili diverse opzioni per la gestione della SVM:

- Fornire l'interfaccia di gestione del cluster nella configurazione backend, insieme alle credenziali appropriate, e specificare il nome SVM.
- Crea un'interfaccia di gestione dedicata per l'SVM utilizzando ONTAP System Manager o la CLI.
- Condividere il ruolo di gestione con un'interfaccia dati NFS.

In ogni caso, l'interfaccia dovrebbe essere in DNS e il nome DNS dovrebbe essere utilizzato durante la configurazione di Trident. Questo aiuta a facilitare alcuni scenari di DR, ad esempio SVM-DR senza l'uso della conservazione dell'identità di rete.

Non esiste una preferenza tra un LIF di gestione dedicato o condiviso per l'SVM, tuttavia, è necessario assicurarsi che le policy di sicurezza della rete siano allineate con l'approccio scelto. In ogni caso, il LIF di gestione dovrebbe essere accessibile tramite DNS per garantire la massima flessibilità qualora "SVM-DR" venga utilizzato in combinazione con Trident.

Limita i volumi totali

I sistemi storage ONTAP hanno un limite massimo di volumi totali, che varia in base alla versione del software e alla piattaforma hardware. Fare riferimento a "NetApp Hardware Universe" per la propria piattaforma specifica e versione di ONTAP per determinare i limiti esatti. Quando i volumi totali sono esauriti, le operazioni di provisioning falliscono non solo per Trident, ma per tutte le richieste di storage.

I driver di Trident `ontap-nas` e `ontap-san` effettuano il provisioning di un FlexVolume per ogni Kubernetes Persistent Volume (PV) creato. Il driver `ontap-nas-economy` crea circa un FlexVolume ogni 200 PV (configurabile tra 50 e 300). Il driver `ontap-san-economy` crea circa un FlexVolume ogni 100 PV (configurabile tra 50 e 200). Per impedire che Trident consumi tutti i volumi disponibili sul sistema storage, è necessario impostare un limite sulla SVM. Puoi farlo dalla riga di comando:

```
vserver modify -vserver <svm_name> -max-volumes <num_of_volumes>
```

Il valore per `max-volumes` varia in base a diversi criteri specifici del tuo ambiente:

- Il numero di volumi esistenti nel cluster ONTAP
- Il numero di volumi che si prevede di effettuare il provisioning al di fuori di Trident per altre applicazioni
- Il numero di volumi persistenti che si prevede saranno consumati dalle applicazioni Kubernetes

Il `max-volumes` valore rappresenta il totale dei volumi forniti su tutti i nodi nel cluster ONTAP, e non su un singolo nodo ONTAP. Di conseguenza, potresti incontrare alcune condizioni in cui un nodo del cluster ONTAP potrebbe avere molti più o meno volumi forniti da Trident rispetto a un altro nodo.

Ad esempio, un cluster ONTAP a due nodi ha la capacità di ospitare un massimo di 2000 volumi FlexVol. Impostare i volumi totali massimi a 1250 sembra molto ragionevole. Tuttavia, se solo "aggregati" di un nodo vengono assegnati alla SVM, o se gli aggregati assegnati da un nodo non possono essere utilizzati per il provisioning (ad esempio, a causa della capacità), allora l'altro nodo diventa la destinazione per tutti i volumi

provisionati da Trident. Questo significa che il limite di volume per quel nodo potrebbe essere raggiunto prima che venga raggiunto il valore `max-volumes`, con un impatto sia su Trident sia sulle altre operazioni sui volumi che utilizzano quel nodo. **Puoi evitare questa situazione assicurandoti che gli aggregati di ciascun nodo del cluster siano assegnati alla SVM utilizzata da Trident in numero uguale.**

Clonare un volume

NetApp Trident supporta la clonazione dei volumi quando si utilizzano i `ontap-nas`, `ontap-san` e `solidfire-san` driver di storage. Quando si utilizzano i `ontap-nas-flexgroup` o `ontap-nas-economy` driver, la clonazione non è supportata. La creazione di un nuovo volume da un volume esistente comporterà la creazione di un nuovo snapshot.

Evitare di clonare un PVC associato a un diverso StorageClass. Eseguire operazioni di clonazione all'interno dello stesso StorageClass per garantire la compatibilità ed evitare comportamenti imprevisti.

Limita la dimensione massima dei volumi creati da Trident

Per configurare la dimensione massima dei volumi che possono essere creati da Trident, utilizzare il `limitVolumeSize` parametro nella `backend.json` definizione.

Oltre a controllare le dimensioni del volume nello storage array, dovresti anche sfruttare le funzionalità di Kubernetes.

Limita la dimensione massima dei FlexVols creati da Trident

Per configurare la dimensione massima per i FlexVols utilizzati come pool per i driver ontap-san-economy e ontap-nas-economy, utilizzare il `limitVolumePoolSize` parametro nella `backend.json` definizione.

Configurare Trident per utilizzare CHAP bidirezionale

È possibile specificare i nomi utente e le password dell'initiator e del target CHAP nella definizione del backend e fare in modo che Trident abiliti CHAP sulla SVM. Utilizzando il parametro `useCHAP` nella configurazione del backend, Trident autentica le connessioni iSCSI per i backend ONTAP con CHAP.

Crea e utilizza una policy QoS SVM

L'utilizzo di una policy QoS ONTAP applicata alla SVM limita il numero di IOPS utilizzabili dai volumi provisionati da Trident. Questo aiuta a "prevenire un bullo" o un container fuori controllo dall'influenzare i carichi di lavoro esterni alla SVM di Trident.

È possibile creare una policy QoS per la SVM in pochi passaggi. Consultare la documentazione per la propria versione di ONTAP per le informazioni più accurate. L'esempio seguente crea una policy QoS che limita il totale di IOPS disponibili per la SVM a 5000.

```
# create the policy group for the SVM
qos policy-group create -policy-group <policy_name> -vserver <svm_name>
-max-throughput 5000iops

# assign the policy group to the SVM, note this will not work
# if volumes or files in the SVM have existing QoS policies
vserver modify -vserver <svm_name> -qos-policy-group <policy_name>
```

Inoltre, se la tua versione di ONTAP lo supporta, puoi valutare l'utilizzo di un QoS minimo per garantire una quantità di throughput ai carichi di lavoro containerizzati. Il QoS adattivo non è compatibile con una policy a livello di SVM.

Il numero di IOPS dedicati ai carichi di lavoro containerizzati dipende da molti aspetti. Tra le altre cose, questi includono:

- Altri carichi di lavoro che utilizzano l'array di storage. Se sono presenti altri carichi di lavoro, non correlati all'implementazione di Kubernetes, che utilizzano le risorse di storage, è necessario prestare attenzione per garantire che tali carichi di lavoro non subiscano accidentalmente impatti negativi.
- Carichi di lavoro previsti in esecuzione nei container. Se carichi di lavoro con elevati requisiti IOPS vengono eseguiti nei container, una policy QoS bassa si traduce in un'esperienza negativa.

È importante ricordare che una policy QoS assegnata a livello di SVM fa sì che tutti i volumi forniti alla SVM condividano lo stesso pool di IOPS. Se una, o un numero limitato, di applicazioni containerizzate ha un elevato requisito di IOPS, potrebbe diventare un problema per gli altri carichi di lavoro containerizzati. Se questo è il caso, potresti voler considerare l'utilizzo di un'automazione esterna per assegnare policy QoS per volume.

Dovresti assegnare il gruppo di policy QoS all'SVM **solo** se la versione di ONTAP è precedente alla 9.8.

Crea gruppi di policy QoS per Trident

La qualità del servizio (QoS) garantisce che le prestazioni dei carichi di lavoro critici non siano degradate da carichi di lavoro concorrenti. I gruppi di policy QoS di ONTAP forniscono opzioni QoS per i volumi e consentono agli utenti di definire il throughput massimo per uno o più carichi di lavoro. Per ulteriori informazioni sulla QoS, fare riferimento a "Garantire il throughput con QoS". È possibile specificare gruppi di policy QoS nel backend o in un pool di storage, e questi vengono applicati a ciascun volume creato in quel pool o backend.

ONTAP dispone di due tipi di gruppi di policy QoS: tradizionali e adattivi. I gruppi di policy tradizionali forniscono un throughput massimo (o minimo, nelle versioni successive) in IOPS. Il QoS adattivo scala automaticamente il throughput in base alle dimensioni del carico di lavoro, mantenendo il rapporto tra IOPS e TB|GB al variare delle dimensioni del carico di lavoro. Questo offre un vantaggio significativo quando si gestiscono centinaia o migliaia di carichi di lavoro in un'implementazione di grandi dimensioni.

Considerare quanto segue quando si creano gruppi di policy QoS:

- Dovresti impostare la `qosPolicy` chiave nel `defaults` blocco della configurazione del backend. Vedi il seguente esempio di configurazione del backend:

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 0.0.0.0
dataLIF: 0.0.0.0
svm: svm0
username: user
password: pass
defaults:
  qosPolicy: standard-pg
storage:
  - labels:
      performance: extreme
    defaults:
      adaptiveQosPolicy: extremely-adaptive-pg
  - labels:
      performance: premium
    defaults:
      qosPolicy: premium-pg
```

- È necessario applicare i gruppi di policy per volume, in modo che ogni volume riceva l'intero throughput come specificato dal gruppo di policy. I gruppi di policy condivisi non sono supportati.

Per ulteriori informazioni sui gruppi di policy QoS, fare riferimento a "Riferimento ai comandi ONTAP".

Limita l'accesso alle risorse di storage ai membri del cluster Kubernetes

Limitare l'accesso ai volumi NFS, alle LUN iSCSI e alle LUN FC create da Trident è un componente fondamentale della strategia di sicurezza per la distribuzione Kubernetes. In questo modo si impedisce agli host che non fanno parte del cluster di accedere ai volumi e di modificare potenzialmente i dati in modo imprevisto.

È importante comprendere che gli spazi dei nomi rappresentano il confine logico per le risorse in Kubernetes. Il presupposto è che le risorse nello stesso spazio dei nomi possano essere condivise, tuttavia, cosa importante, non esiste alcuna funzionalità cross-namespace. Ciò significa che, sebbene i PV siano oggetti globali, quando associati a un PVC sono accessibili solo ai pod che si trovano nello stesso spazio dei nomi. **È fondamentale garantire che gli spazi dei nomi vengano utilizzati per fornire la separazione quando appropriato.**

La preoccupazione principale per la maggior parte delle organizzazioni in merito alla sicurezza dei dati in un contesto Kubernetes è che un processo in un container possa accedere a storage montato sull'host, ma non destinato al container. "Spazi dei nomi" sono progettati per prevenire questo tipo di compromissione. Tuttavia, esiste un'eccezione: i container privilegiati.

Un container privilegiato è un container che viene eseguito con autorizzazioni a livello di host notevolmente superiori al normale. Queste non vengono negate per impostazione predefinita, quindi assicurati di disabilitare la funzionalità utilizzando "politiche di sicurezza del pod".

Per i volumi in cui è richiesto l'accesso sia da Kubernetes che da host esterni, lo storage dovrebbe essere gestito in modo tradizionale, con il PV introdotto dall'amministratore e non gestito da Trident. Questo garantisce che il volume di storage venga distrutto solo quando sia Kubernetes che gli host esterni si sono

disconnessi e non utilizzano più il volume. Inoltre, è possibile applicare una policy di esportazione personalizzata, che consente l'accesso dai nodi del cluster Kubernetes e dai server di destinazione esterni al cluster.

Per le distribuzioni che dispongono di nodi infrastrutturali dedicati (ad esempio, OpenShift) o altri nodi che non sono in grado di pianificare le applicazioni utente, è necessario utilizzare policy di esportazione separate per limitare ulteriormente l'accesso alle risorse di storage. Ciò include la creazione di una policy di esportazione per i servizi che sono distribuiti su tali nodi infrastrutturali (ad esempio, i servizi di Metriche e Logging di OpenShift), e per le applicazioni standard che sono distribuite su nodi non infrastrutturali.

Utilizzare una policy di esportazione dedicata

È necessario assicurarsi che esista una policy di esportazione per ogni backend che consenta l'accesso solo ai nodi presenti nel cluster Kubernetes. Trident può creare e gestire automaticamente le policy di esportazione. In questo modo, Trident limita l'accesso ai volumi che fornisce ai nodi nel cluster Kubernetes e semplifica l'aggiunta/eliminazione di nodi.

In alternativa, puoi anche creare manualmente una policy di esportazione e popolarla con una o più regole di esportazione che elaborano ogni richiesta di accesso al nodo:

- Utilizzare il comando ONTAP CLI `vserver export-policy create` per creare la policy di esportazione.
- Aggiungere regole alla policy di esportazione utilizzando il `vserver export-policy rule create` comando ONTAP CLI.

L'esecuzione di questi comandi consente di limitare quali nodi Kubernetes hanno accesso ai dati.

Disabilita `showmount` per l'applicazione SVM

La `showmount`funzionalità consente a un client NFS di interrogare l'SVM per un elenco delle esportazioni NFS disponibili. Un pod distribuito nel cluster Kubernetes può emettere il comando `showmount -e` contro la SVM e ricevere un elenco dei mount disponibili, inclusi quelli a cui non ha accesso. Sebbene questo, di per sé, non costituisca una compromissione della sicurezza, fornisce informazioni non necessarie che potrebbero aiutare un utente non autorizzato a connettersi a un'esportazione NFS.

È necessario disabilitare `showmount` utilizzando il comando CLI di ONTAP a livello SVM:

```
vserver nfs modify -vserver <svm_name> -showmount disabled
```

Best practice per SolidFire

Scopri le best practice per la configurazione dello storage SolidFire per Trident.

Crea account SolidFire

Ogni SolidFire account rappresenta un proprietario univoco del volume e riceve il proprio set di credenziali Challenge-Handshake Authentication Protocol (CHAP). Puoi accedere ai volumi assegnati a un account utilizzando il nome dell'account e le relative credenziali CHAP oppure tramite un gruppo di accesso al volume. A un account possono essere assegnati fino a duemila volumi, ma un volume può appartenere a un solo account.

Creare una policy QoS

Utilizzare le policy di Quality of Service (QoS) di SolidFire se si desidera creare e salvare un'impostazione di qualità del servizio standardizzata che può essere applicata a molti volumi.

È possibile impostare i parametri QoS su base per-volume. Le prestazioni per ogni volume possono essere garantite impostando tre parametri configurabili che definiscono la QoS: Min IOPS, Max IOPS e Burst IOPS.

Ecco i possibili valori minimi, massimi e burst di IOPS per la dimensione del blocco da 4Kb.

Parametro IOPS	Definizione	Valore minimo	Valore predefinito	Valore massimo(4Kb)
IOPS minimi	Il livello garantito di prestazioni per un volume.	50	50	15000
IOPS massimi	Le prestazioni non supereranno questo limite.	50	15000	200.000
IOPS a raffica	IOPS massimi consentiti in uno scenario di burst breve.	50	15000	200.000

Sebbene i valori Max IOPS e Burst IOPS possano essere impostati fino a 200.000, le prestazioni massime reali di un volume sono limitate dall'utilizzo del cluster e dalle prestazioni per nodo.

La dimensione dei blocchi e la larghezza di banda influiscono direttamente sul numero di IOPS. All'aumentare delle dimensioni dei blocchi, il sistema aumenta la larghezza di banda fino al livello necessario per elaborare blocchi di dimensioni maggiori. All'aumentare della larghezza di banda, il numero di IOPS che il sistema è in grado di raggiungere diminuisce. Fare riferimento a "Quality of Service SolidFire" per ulteriori informazioni su QoS e prestazioni.

Autenticazione SolidFire

Element supporta due metodi di autenticazione: CHAP e Volume Access Groups (VAG). CHAP utilizza il protocollo CHAP per autenticare l'host al backend. Volume Access Groups controlla l'accesso ai volumi che mette a disposizione. NetApp consiglia di utilizzare CHAP per l'autenticazione in quanto è più semplice e non ha limiti di scala.

Trident con il provisioner CSI avanzato supporta l'uso dell'autenticazione CHAP. I VAG devono essere utilizzati solo nella modalità operativa tradizionale non CSI.

L'autenticazione CHAP (verifica che l'initiator sia l'utente previsto del volume) è supportata solo con il controllo dell'accesso basato sull'account. Se si utilizza CHAP per l'autenticazione, sono disponibili due opzioni: CHAP unidirezionale e CHAP bidirezionale. La CHAP unidirezionale autentica l'accesso al volume utilizzando il nome dell'account SolidFire e il segreto dell'initiator. L'opzione CHAP bidirezionale offre il modo più sicuro di autenticare il volume, poiché il volume autentica l'host tramite il nome dell'account e il segreto dell'initiator, e quindi l'host autentica il volume tramite il nome dell'account e il segreto della destinazione.

Tuttavia, se non è possibile abilitare CHAP e sono necessari i VAG, crea il gruppo di accesso e aggiungi gli initiator host e i volumi al gruppo di accesso. Ogni IQN che aggiungi a un gruppo di accesso può accedere a ogni volume del gruppo con o senza autenticazione CHAP. Se l'iSCSI initiator è configurato per usare l'autenticazione CHAP, viene utilizzato il controllo di accesso basato sull'account. Se l'iSCSI initiator non è configurato per usare l'autenticazione CHAP, viene utilizzato il controllo di accesso Volume Access Group.

Dove trovare ulteriori informazioni?

Di seguito sono elencate alcune delle documentazioni relative alle best practice. Cerca la "Libreria NetApp" per le versioni più aggiornate.

ONTAP

- "NFS Guida alle migliori pratiche e all'implementazione"
- "Amministrazione SAN" (per iSCSI)
- "Configurazione rapida iSCSI per RHEL"

Software Element

- "Configurazione di SolidFire per Linux"

NetApp HCI

- "Prerequisiti per l'implementazione di NetApp HCI"
- "Accedere al NetApp Deployment Engine"

Informazioni sulle best practice applicative

- "Best practice per MySQL su ONTAP"
- "Le migliori pratiche per MySQL su SolidFire"
- "NetApp SolidFire e Cassandra"
- "Best practice Oracle su SolidFire"
- "Best practice PostgreSQL su SolidFire"

Non tutte le applicazioni hanno linee guida specifiche, è importante collaborare con il tuo team NetApp e utilizzare il "Libreria NetApp" per trovare la documentazione più aggiornata.

Integra Trident

Per integrare Trident, è necessario integrare i seguenti elementi di progettazione e architettura: selezione e distribuzione del driver, progettazione della classe di archiviazione, progettazione del pool virtuale, impatto del Persistent Volume Claim (PVC) sul provisioning dello storage, operazioni sui volumi e distribuzione dei servizi OpenShift tramite Trident.

Selezione e distribuzione del driver

Seleziona e distribuisci un driver backend per il tuo sistema storage.

Driver backend ONTAP

I driver backend ONTAP si differenziano in base al protocollo utilizzato e al modo in cui i volumi vengono forniti sul sistema storage. Pertanto, valuta attentamente quale driver implementare.

A un livello superiore, se la tua applicazione ha componenti che necessitano di storage condiviso (più pod che accedono allo stesso PVC), i driver basati su NAS rappresentano la scelta predefinita, mentre i driver iSCSI basati su blocchi soddisfano le esigenze di storage non condiviso. Scegli il protocollo in base ai requisiti dell'applicazione e al livello di comfort dei team di storage e infrastruttura. In generale, ci sono poche differenze tra loro per la maggior parte delle applicazioni, quindi spesso la decisione si basa sul fatto che sia necessario o meno uno storage condiviso (dove più di un pod avrà bisogno di accesso simultaneo).

I driver backend ONTAP disponibili sono:

- `ontap-nas`: Ogni PV fornito è un ONTAP FlexVolume completo.
- `ontap-nas-economy`: Ogni PV fornito è un qtree, con un numero configurabile di qtree per FlexVolume (il valore predefinito è 200).
- `ontap-nas-flexgroup`: Ogni PV è predisposto come un ONTAP FlexGroup completo e vengono utilizzati tutti gli aggregati assegnati a una SVM.
- `ontap-san`: Ogni PV fornito è una LUN all'interno del proprio FlexVolume.
- `ontap-san-economy`: Ogni PV fornito è una LUN, con un numero configurabile di LUN per FlexVolume (il valore predefinito è 100).

La scelta tra i tre driver NAS ha alcune ramificazioni sulle funzionalità rese disponibili all'applicazione.

Si noti che, nelle tabelle seguenti, non tutte le funzionalità sono esposte tramite Trident. Alcune devono essere applicate dall'amministratore dello storage dopo il provisioning se si desidera quella funzionalità. Le note a piè di pagina in apice distinguono la funzionalità per funzionalità e driver.

Driver NAS ONTAP	Istantanee	Cloni	Politiche di esportazione dinamiche	Multi-attach	QoS	Ridimensiona	Replica
`ontap-nas`	Sì	Sì	Sì [5]	Sì	Sì [1]	Sì	Sì [1]
`ontap-nas-economy`	NO [3]	NO [3]	Sì [5]	Sì	NO [3]	Sì	NO [3]
`ontap-nas-flexgroup`	Sì [1]	NO	Sì [5]	Sì	Sì [1]	Sì	Sì [1]

Trident offre 2 driver SAN per ONTAP, le cui funzionalità sono mostrate di seguito.

Driver SAN ONTAP	Istantanee	Cloni	Multi-attach	CHAP bidirezionale	QoS	Ridimensiona	Replica
`ontap-san`	Sì	Sì	Sì [4]	Sì	Sì [1]	Sì	Sì [1]
`ontap-san-economy`	Sì	Sì	Sì [4]	Sì	NO [3]	Sì	NO [3]

> Nota a piè di pagina per le tabelle precedenti: Sì [1]: Non gestito da Trident Sì [2]: Gestito da Trident, ma non PV granulare NO [3]: Non gestito da Trident e non PV granulare Sì [4]: Supportato per volumi raw-block Sì [5]: Supportato da Trident

Le funzionalità che non sono granulari al PV vengono applicate all'intero FlexVolume e tutti i PV (cioè qtrees o LUN in FlexVols condivisi) condivideranno una pianificazione comune.

Come si può vedere nelle tabelle precedenti, gran parte della funzionalità tra il `ontap-nas` e il `ontap-nas-economy` è la stessa. Tuttavia, poiché il driver `ontap-nas-economy` limita la possibilità di controllare la pianificazione a granularità per-PV, ciò può influire in particolare sulla pianificazione del disaster recovery e del backup. Per i team di sviluppo che desiderano sfruttare la funzionalità di clonazione PVC sullo storage ONTAP, ciò è possibile solo quando si utilizzano i driver `ontap-nas`, `ontap-san` o `ontap-san-economy`.

Il `solidfire-san` driver è anche in grado di clonare i PVC.

Driver di backend Cloud Volumes ONTAP

Cloud Volumes ONTAP fornisce il controllo dei dati insieme a funzionalità di storage di classe enterprise per vari casi d'uso, inclusi file share e storage a livello di blocco che servono protocolli NAS e SAN (NFS, SMB / CIFS e iSCSI). I driver compatibili per Cloud Volumes ONTAP sono `ontap-nas`, `ontap-nas-economy`, `ontap-san` e `ontap-san-economy`. Questi sono applicabili per Cloud Volumes ONTAP per Azure, Cloud Volumes ONTAP per GCP.

Driver di backend Amazon FSx for ONTAP

Amazon FSx for NetApp ONTAP consente di sfruttare le funzionalità, le prestazioni e le capacità amministrative di NetApp già note, sfruttando al contempo la semplicità, l'agilità, la sicurezza e la scalabilità dell'archiviazione dei dati su AWS. FSx for ONTAP supporta molte funzionalità del file system ONTAP e API di amministrazione. I driver compatibili per Cloud Volume ONTAP sono `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `ontap-san` e `ontap-san-economy`.

NetApp HCI/SolidFire driver backend

Il `solidfire-san` driver utilizzato con le piattaforme NetApp HCI/SolidFire aiuta l'amministratore a configurare un backend Element per Trident sulla base di limiti QoS. Se si desidera progettare il backend per impostare limiti QoS specifici sui volumi forniti da Trident, utilizzare il parametro `type` nel file di backend. L'amministratore può anche limitare la dimensione del volume che può essere creato sullo storage utilizzando il parametro `limitVolumeSize`. Attualmente, le funzionalità di Element storage come il ridimensionamento del volume e la replica del volume non sono supportate tramite il `solidfire-san` driver. Queste operazioni devono essere eseguite manualmente tramite l'interfaccia web di Element Software.

Driver SolidFire	Istantanee	Cloni	Multi-attach	CHAP	QoS	Ridimensiona	Replica
`solidfire-san`	Sì	Sì	Sì [2]	Sì	Sì	Sì	Sì [1]

> Nota a piè di pagina: Sì [1]: Non gestito da Trident Sì [2]: Supportato per volumi raw-block

Driver backend di Azure NetApp Files

Trident utilizza il `azure-netapp-files` driver per gestire il servizio "Azure NetApp Files".

Ulteriori informazioni su questo driver e su come configurarlo sono disponibili in "Configurazione del backend Trident per Azure NetApp Files".

Driver Azure NetApp Files	Istantanee	Cloni	Multi-attach	QoS	Espandi	Replica
`azure-netapp-files`	Sì	Sì	Sì	Sì	Sì	Sì [1]

> Nota a piè di pagina: Sì [1]: Non gestito da Trident

Progettazione della storage class

Le singole Storage class devono essere configurate e applicate per creare un oggetto Kubernetes Storage Class. Questa sezione illustra come progettare una storage class per la tua applicazione.

Utilizzo specifico del backend

Il filtraggio può essere utilizzato all'interno di uno specifico oggetto della storage class per determinare quale pool di storage o insieme di pool deve essere utilizzato con quella specifica storage class. Tre serie di filtri possono essere impostate nella Storage Class: `storagePools`, `additionalStoragePools`, e/o `excludeStoragePools`.

Il `storagePools` parametro consente di limitare lo storage all'insieme di pool che corrispondono agli attributi specificati. Il `additionalStoragePools` parametro viene utilizzato per estendere l'insieme di pool che Trident utilizza per il provisioning insieme all'insieme di pool selezionati dagli attributi e dai parametri `storagePools`. È possibile utilizzare uno dei due parametri da solo o entrambi insieme per assicurarsi che venga selezionato l'insieme appropriato di pool di storage.

Il `excludeStoragePools` parametro viene utilizzato per escludere specificamente l'insieme elencato di pool che corrispondono agli attributi.

Emula le policy QoS

Se si desidera progettare Storage Classes per emulare le politiche di Quality of Service, creare una Storage Class con l'attributo `media` come `hdd` o `ssd`. In base all'attributo `media` menzionato nella Storage Class, Trident selezionerà il backend appropriato che serve `hdd` o `ssd` aggregati per corrispondere all'attributo media e quindi dirigerà il provisioning dei volumi sull'aggregato specifico. Pertanto, è possibile creare una Storage Class PREMIUM con l'attributo `media` impostato come `ssd` che potrebbe essere classificata come la politica QoS PREMIUM. È possibile creare un'altra Storage Class STANDARD con l'attributo media impostato come `hdd' che potrebbe essere classificata come la politica QoS STANDARD. Si può anche utilizzare l'attributo ``IOPS'' nella Storage Class per reindirizzare il provisioning a un'appliance Element che può essere definita come una politica QoS.

Utilizzare il backend basato su funzionalità specifiche

Le classi di storage possono essere progettate per indirizzare il provisioning dei volumi su un backend specifico in cui sono abilitate funzionalità quali thin e thick provisioning, snapshot, cloni e crittografia. Per specificare quale storage utilizzare, crea Storage Classes che specificano il backend appropriato con la funzionalità richiesta abilitata.

Pool virtuali

I pool virtuali sono disponibili per tutti i backend Trident. È possibile definire pool virtuali per qualsiasi backend, utilizzando qualsiasi driver che Trident fornisce.

I pool virtuali consentono a un amministratore di creare un livello di astrazione sui backend che possono essere referenziati attraverso le Storage Classes, per una maggiore flessibilità e un posizionamento efficiente dei volumi sui backend. Backend diversi possono essere definiti con la stessa classe di servizio. Inoltre, è possibile creare più pool di storage sullo stesso backend ma con caratteristiche diverse. Quando una Storage Class è configurata con un selettore con etichette specifiche, Trident sceglie un backend che corrisponde a tutte le etichette del selettore per posizionare il volume. Se le etichette del selettore della Storage Class corrispondono a più pool di storage, Trident sceglierà uno di essi da cui effettuare il provisioning del volume.

Progettazione di pool virtuali

Durante la creazione di un backend, è generalmente possibile specificare un set di parametri. Era impossibile per l'amministratore creare un altro backend con le stesse credenziali di storage e con un set di parametri diverso. Con l'introduzione dei pool virtuali, questo problema è stato alleviato. Un pool virtuale è un'astrazione di livello introdotta tra il backend e la Kubernetes Storage Class, in modo che l'amministratore possa definire parametri insieme a etichette che possono essere referenziate tramite le Kubernetes Storage Classes come selettore, in modo backend-agnostic. I pool virtuali possono essere definiti per tutti i backend NetApp supportati con Trident. Questo elenco include SolidFire/NetApp HCI, ONTAP, così come Azure NetApp Files.

> Quando si definiscono pool virtuali, si consiglia di non tentare di riorganizzare l'ordine dei pool virtuali esistenti in una definizione di backend. Si consiglia inoltre di non modificare/modificare gli attributi di un pool virtuale esistente e di definire invece un nuovo pool virtuale.

Emulazione di diversi livelli di servizio/QoS

È possibile progettare pool virtuali per emulare classi di servizio. Utilizzando l'implementazione del pool virtuale per Cloud Volume Service for Azure NetApp Files, esaminiamo come possiamo configurare diverse classi di servizio. Configura il backend Azure NetApp Files con più etichette, che rappresentano diversi livelli di prestazioni. Imposta `servicelevel` l'aspetto sul livello di prestazioni appropriato e aggiungi altri aspetti richiesti sotto ciascuna etichetta. Ora crea diverse Kubernetes Storage Classes che verranno mappate a diversi pool virtuali. Utilizzando il campo `parameters.selector`, ogni StorageClass indica quali pool virtuali possono essere utilizzati per ospitare un volume.

Assegnazione di un insieme specifico di aspetti

È possibile progettare più pool virtuali con un set specifico di aspetti da un singolo backend di storage. Per farlo, configura il backend con più etichette e imposta gli aspetti richiesti sotto ciascuna etichetta. Ora crea diverse classi di storage Kubernetes utilizzando il campo `parameters.selector` che verranno mappate a diversi pool virtuali. I volumi che vengono provisionati sul backend avranno gli aspetti definiti nel pool virtuale scelto.

Caratteristiche del PVC che influenzano il provisioning dello storage

Alcuni parametri oltre la storage class richiesta possono influire sul processo decisionale di provisioning di Trident durante la creazione di un PVC.

Modalità di accesso

Quando si richiede storage tramite PVC, uno dei campi obbligatori è la modalità di accesso. La modalità desiderata può influire sul backend selezionato per ospitare la richiesta di storage.

Trident tenterà di abbinare il protocollo storage utilizzato al metodo di accesso specificato in base alla seguente matrice. Questo è indipendente dalla piattaforma di storage sottostante.

	ReadWriteOnce	ReadOnlyMany	ReadWriteMany
iSCSI	Sì	Sì	Sì (blocco raw)
NFS	Sì	Sì	Sì

Una richiesta per un ReadWriteMany PVC inviata a una distribuzione Trident senza un backend NFS configurato non comporterà il provisioning di alcun volume. Per questo motivo, il richiedente dovrebbe utilizzare la modalità di accesso appropriata per la propria applicazione.

Operazioni volume

Modificare i volumi persistenti

I volumi persistenti sono, con due eccezioni, oggetti immutabili in Kubernetes. Una volta creati, la reclaim policy e le dimensioni possono essere modificate. Tuttavia, ciò non impedisce che alcuni aspetti del volume vengano modificati al di fuori di Kubernetes. Questo può essere utile per personalizzare il volume per applicazioni specifiche, per garantire che la capacità non venga consumata accidentalmente o semplicemente per spostare il volume su un altro storage controller per qualsiasi motivo.

> I provisioner in-tree di Kubernetes al momento non supportano le operazioni di ridimensionamento dei volumi per PV NFS, iSCSI o FC. Trident supporta l'espansione dei volumi NFS, iSCSI e FC.

I dettagli di connessione del PV non possono essere modificati dopo la creazione.

Crea snapshot dei volumi su richiesta

Trident supporta la creazione di snapshot di volume on-demand e la creazione di PVC da snapshot utilizzando il framework CSI. Gli snapshot forniscono un metodo pratico per mantenere copie point-in-time dei dati e hanno un ciclo di vita indipendente dal PV sorgente in Kubernetes. Questi snapshot possono essere utilizzati per clonare i PVC.

Crea volumi da snapshot

Trident supporta anche la creazione di PersistentVolumes da snapshot di volume. Per farlo, è sufficiente creare un PersistentVolumeClaim e specificare il `datasource` come snapshot richiesto da cui deve essere creato il volume. Trident gestirà questo PVC creando un volume con i dati presenti nello snapshot. Con questa funzionalità, è possibile duplicare i dati tra regioni, creare ambienti di test, sostituire completamente un volume di produzione danneggiato o corrotto, oppure recuperare file e directory specifici e trasferirli su un altro volume collegato.

Sposta i volumi nel cluster

Gli amministratori di storage hanno la possibilità di spostare volumi tra aggregati e controller nel cluster ONTAP senza interruzioni per il consumatore di storage. Questa operazione non influisce su Trident o sul cluster Kubernetes, purché l'aggregato di destinazione sia uno a cui l'SVM che Trident sta utilizzando ha accesso. È importante sottolineare che, se l'aggregato è stato appena aggiunto all'SVM, il backend dovrà essere aggiornato aggiungendolo nuovamente a Trident. Questo farà sì che Trident reinventari l'SVM in modo che il nuovo aggregato venga riconosciuto.

Tuttavia, lo spostamento di volumi tra backend non è supportato automaticamente da Trident. Questo include lo spostamento tra SVM nello stesso cluster, tra cluster o su una diversa piattaforma di storage (anche se tale sistema storage è connesso a Trident).

Se un volume viene copiato in un'altra posizione, la funzionalità di importazione del volume può essere utilizzata per importare i volumi correnti in Trident.

Espandi volumi

Trident supporta il ridimensionamento dei volumi permanenti NFS, iSCSI e FC. Ciò consente agli utenti di ridimensionare i propri volumi direttamente tramite il livello Kubernetes. L'espansione del volume è possibile per tutte le principali piattaforme di storage NetApp, inclusi ONTAP e backend SolidFire/NetApp HCI. Per consentire un'eventuale espansione successiva, impostare `allowVolumeExpansion` su `true` nel StorageClass associato al volume. Ogni volta che è necessario ridimensionare il volume permanente, modificare l'annotazione `spec.resources.requests.storage` nella richiesta del volume permanente con la dimensione del volume richiesta. Trident si occuperà automaticamente del ridimensionamento del volume sul cluster.

Importa un volume esistente in Kubernetes

L'importazione di volumi offre la possibilità di importare un volume di storage esistente in un ambiente Kubernetes. Questa funzionalità è attualmente supportata dai `ontap-nas`, `ontap-nas-flexgroup`, `solidfire-san` e `azure-netapp-files` driver. Questa funzionalità è utile quando si esegue il porting di un'applicazione esistente in Kubernetes o durante scenari di disaster recovery.

Quando si utilizzano i driver ONTAP e `solidfire-san`, utilizzare il comando `tridentctl import volume <backend-name> <volume-name> -f /path/pvc.yaml` per importare un volume esistente in Kubernetes da gestire tramite Trident. Il file PVC YAML o JSON utilizzato nel comando di importazione del volume punta a una storage class che identifica Trident come provisioner. Quando si utilizza un backend NetApp HCI/SolidFire, assicurarsi che i nomi dei volumi siano univoci. Se i nomi dei volumi sono duplicati, clonare il volume con un nome univoco in modo che la funzionalità di importazione dei volumi possa distinguerli.

Se si utilizza il `azure-netapp-files` driver, utilizzare il comando `tridentctl import volume <backend-name> <volume path> -f /path/pvc.yaml` per importare il volume in Kubernetes affinché venga gestito da Trident. Ciò garantisce un riferimento univoco al volume.

Quando il comando sopra riportato viene eseguito, Trident troverà il volume sul backend e ne leggerà la dimensione. Aggiungerà automaticamente (e sovrascriverà se necessario) la dimensione del volume del PVC configurato. Trident crea quindi il nuovo PV e Kubernetes associa il PVC al PV.

Se un container è stato distribuito in modo da richiedere lo specifico PVC importato, rimarrà in sospeso finché la coppia PVC/PV non viene associata tramite il processo di importazione del volume. Dopo che la coppia PVC/PV è stata associata, il container dovrebbe avviarsi, a condizione che non ci siano altri problemi.

Servizio di registro

L'implementazione e la gestione dello storage per il registry sono state documentate su "netapp.io" nel "blog".

Servizio di logging

Come altri servizi OpenShift, il servizio di logging viene distribuito tramite Ansible con parametri di configurazione forniti dal file di inventario, ovvero hosts, forniti al playbook. Verranno trattati due metodi di installazione: la distribuzione del logging durante l'installazione iniziale di OpenShift e la distribuzione del

logging dopo che OpenShift è stato installato.

> A partire dalla versione 3.9 di Red Hat OpenShift, la documentazione ufficiale sconsiglia l'utilizzo di NFS per il servizio di logging a causa di problemi di corruzione dei dati. Questo si basa sui test effettuati da Red Hat sui propri prodotti. Il server NFS ONTAP non presenta questi problemi e può facilmente supportare un'implementazione di logging. In definitiva, la scelta del protocollo per il servizio di logging spetta a te, sappi solo che entrambi funzioneranno perfettamente utilizzando piattaforme NetApp e non c'è motivo di evitare NFS se questa è la tua preferenza.

Se si sceglie di utilizzare NFS con il servizio di registrazione, sarà necessario impostare la variabile Ansible `openshift_enable_unsupported_configurations` su `true` per impedire il fallimento del programma di installazione.

Inizia

Il servizio di logging può, facoltativamente, essere distribuito sia per le applicazioni sia per le operazioni principali del OpenShift cluster stesso. Se si sceglie di distribuire il logging delle operazioni, specificando la variabile `openshift_logging_use_ops` come `true`, verranno create due istanze del servizio. Le variabili che controllano l'istanza di logging per le operazioni contengono "ops", mentre l'istanza per le applicazioni no.

Configurare le variabili Ansible in base al metodo di distribuzione è importante per garantire che lo storage corretto sia utilizzato dai servizi sottostanti. Diamo un'occhiata alle opzioni per ciascun metodo di distribuzione.

> Le tabelle seguenti contengono solo le variabili rilevanti per la configurazione dello storage in relazione al servizio di logging. Puoi trovare altre opzioni in "Documentazione di logging di Red Hat OpenShift" che dovrebbero essere riviste, configurate e utilizzate in base alla tua distribuzione.

Le variabili riportate nella tabella seguente porteranno il playbook di Ansible a creare un PV e un PVC per il servizio di logging utilizzando i dettagli forniti. Questo metodo è significativamente meno flessibile rispetto all'utilizzo del playbook di installazione dei componenti dopo l'installazione di OpenShift, tuttavia, se si dispone di volumi esistenti, è un'opzione.

Variabile	Dettagli
`openshift_logging_storage_kind`	Impostare `nfs` per fare in modo che il programma di installazione crei un PV NFS per il servizio di logging.
`openshift_logging_storage_host`	Il nome host o l'indirizzo IP dell'host NFS. Questo deve essere impostato sul dataLIF per la tua macchina virtuale.
`openshift_logging_storage_nfs_directory`	Il percorso di montaggio per l'esportazione NFS. Ad esempio, se il volume è giuntato come `/openshift_logging`, si dovrebbe usare quel percorso per questa variabile.
`openshift_logging_storage_volume_name`	Il nome, ad esempio `pv_ose_logs`, del PV da creare.
`openshift_logging_storage_volume_size`	La dimensione dell'esportazione NFS, ad esempio `100Gi`.

Se il OpenShift cluster è già in esecuzione e quindi Trident è stato distribuito e configurato, il programma di

installazione può utilizzare il provisioning dinamico per creare i volumi. Le seguenti variabili dovranno essere configurate.

Variabile	Dettagli
`openshift_logging_es_pvc_dynamic`	Impostare su true per utilizzare volumi con provisioning dinamico.
`openshift_logging_es_pvc_storage_class_name`	Il nome della storage class che verrà utilizzata nel PVC.
`openshift_logging_es_pvc_size`	La dimensione del volume richiesto nel PVC.
`openshift_logging_es_pvc_prefix`	Un prefisso per i PVC utilizzati dal servizio di logging.
`openshift_logging_es_ops_pvc_dynamic`	Impostare su `true` per utilizzare volumi forniti dinamicamente per l'istanza di logging ops.
`openshift_logging_es_ops_pvc_storage_class_name`	Il nome della storage class per l'istanza di ops logging.
`openshift_logging_es_ops_pvc_size`	La dimensione della richiesta di volume per l'istanza ops.
`openshift_logging_es_ops_pvc_prefix`	Un prefisso per i PVC dell'istanza ops.

Distribuire lo stack di registrazione

Se si sta distribuendo il logging come parte del processo di installazione iniziale di OpenShift, è sufficiente seguire il processo di distribuzione standard. Ansible configurerà e distribuirà i servizi necessari e gli oggetti OpenShift in modo che il servizio sia disponibile non appena Ansible completa.

Tuttavia, se si esegue il deploy dopo l'installazione iniziale, il playbook del componente dovrà essere usato da Ansible. Questa procedura può cambiare leggermente con versioni diverse di OpenShift, quindi assicurati di leggere e seguire "Red Hat OpenShift Container Platform 3.11 documentazione" per la tua versione.

Servizio metriche

Il servizio metriche fornisce all'amministratore informazioni preziose sullo stato, l'utilizzo delle risorse e la disponibilità del OpenShift cluster. È inoltre necessario per la funzionalità di auto-scale del pod e molte organizzazioni utilizzano i dati del servizio metriche per le applicazioni di charge back e/o show back.

Come per il servizio di logging e OpenShift nel suo complesso, Ansible viene usato per distribuire il servizio metriche. Inoltre, come il servizio di logging, il servizio metriche può essere distribuito durante la configurazione iniziale del cluster o dopo che è operativo utilizzando il metodo di installazione dei componenti. Le tabelle seguenti contengono le variabili importanti quando si configura storage persistente per il servizio metriche.

Le tabelle seguenti contengono solo le variabili rilevanti per la configurazione dello storage in relazione al servizio metriche. Ci sono molte altre opzioni che si trovano nella documentazione e che devono essere riviste, configurate e utilizzate in base alla propria distribuzione.

Variabile	Dettagli
`openshift_metrics_storage_kind`	Impostare `nfs` per fare in modo che il programma di installazione crei un PV NFS per il servizio di logging.

Variabile	Dettagli
`openshift_metrics_storage_host`	Il nome host o l'indirizzo IP dell'host NFS. Questo deve essere impostato sul dataLIF per il tuo SVM.
`openshift_metrics_storage_nfs_directory`	Il percorso di montaggio per l'esportazione NFS. Ad esempio, se il volume è giuntato come `/openshift_metrics`, si dovrebbe usare quel percorso per questa variabile.
`openshift_metrics_storage_volume_name`	Il nome, ad esempio `pv_ose_metrics`, del PV da creare.
`openshift_metrics_storage_volume_size`	La dimensione dell'esportazione NFS, ad esempio `100Gi`.

Se il OpenShift cluster è già in esecuzione e quindi Trident è stato distribuito e configurato, il programma di installazione può utilizzare il provisioning dinamico per creare i volumi. Le seguenti variabili dovranno essere configurate.

Variabile	Dettagli
`openshift_metrics_cassandra_pvc_prefix`	Un prefisso da utilizzare per le metriche PVC.
`openshift_metrics_cassandra_pvc_size`	La dimensione dei volumi da richiedere.
`openshift_metrics_cassandra_storage_type`	Il tipo di storage da utilizzare per le metriche, deve essere impostato su dinamico affinché Ansible crei i PVC con la classe di storage appropriata.
`openshift_metrics_cassanda_pvc_storage_class_name`	Il nome della storage class da utilizzare.

Distribuire il servizio metriche

Con le variabili Ansible appropriate definite nel file hosts/inventory, distribuisci il servizio utilizzando Ansible. Se stai distribuendo al momento dell'installazione di OpenShift, allora il PV verrà creato e utilizzato automaticamente. Se stai distribuendo utilizzando i playbook dei componenti, dopo l'installazione di OpenShift, allora Ansible crea tutti i PVC necessari e, dopo che Trident ha eseguito il provisioning dello storage per essi, distribuisce il servizio.

Le variabili di cui sopra e il processo di distribuzione possono cambiare con ogni versione di OpenShift. Assicuratevi di rivedere e seguire "Guida alla distribuzione di OpenShift di Red Hat" per la vostra versione affinché sia configurato per il vostro ambiente.

Protezione dei dati e disaster recovery

Scoprite le opzioni di protezione e recovery per Trident e i volumi creati utilizzando Trident. Dovreste avere una strategia di protezione e recovery dei dati per ogni applicazione con un requisito di persistenza.

Replica e recovery di Trident

È possibile creare un backup per ripristinare Trident in caso di disastro.

Replica di Trident

Trident utilizza i CRD di Kubernetes per memorizzare e gestire il proprio stato e il cluster Kubernetes etcd per memorizzare i suoi metadati.

Passaggi

1. Eseguire il backup del cluster Kubernetes etcd utilizzando "Kubernetes: Backup di un cluster etcd".
2. Posiziona gli artefatti di backup su un volume FlexVol

> NetApp consiglia di proteggere l'SVM in cui risiede il FlexVol con una relazione di SnapMirror con un altro SVM.

Recovery di Trident

Utilizzando i CRD Kubernetes e lo snapshot etcd del cluster Kubernetes, è possibile recuperare Trident.

Passaggi

1. Dal SVM di destinazione, montare il volume che contiene i file di dati etcd di Kubernetes e i certificati sull'host che sarà configurato come nodo master.
2. Copia tutti i certificati richiesti relativi al cluster Kubernetes sotto `/etc/kubernetes/pki` e i file dei membri etcd sotto `/var/lib/etcd`.
3. Ripristina il cluster Kubernetes dal backup etcd utilizzando "Kubernetes: Ripristino di un cluster etcd".
4. Eseguire `kubectl get crd` per verificare che tutte le risorse personalizzate di Trident siano state avviate e recuperare gli oggetti di Trident per verificare che tutti i dati siano disponibili.

Replica e recovery SVM

Trident non è in grado di configurare le relazioni di replica, tuttavia l'amministratore dello storage può utilizzare "ONTAP SnapMirror" per replicare un SVM.

In caso di disastro, è possibile attivare la SnapMirror destination SVM per iniziare a servire i dati. È possibile tornare al primario quando i sistemi vengono ripristinati.

Informazioni su questa attività

Considerare quanto segue quando si utilizza la funzionalità di replica SVM SnapMirror:

- Dovresti creare un backend distinto per ogni SVM con SVM-DR abilitato.
- Configura le classi di storage per selezionare i backend replicati solo quando necessario, per evitare che i volumi che non necessitano di replica vengano forniti sui backend che supportano SVM-DR.
- Gli amministratori delle applicazioni devono comprendere i costi aggiuntivi e la complessità associati alla replica e valutare attentamente il proprio piano di recovery prima di iniziare questo processo.

Replicazione SVM

È possibile utilizzare "ONTAP: SnapMirror SVM replicazione" per creare la relazione di replica SVM.

SnapMirror consente di impostare opzioni per controllare cosa replicare. Dovrai sapere quali opzioni hai selezionato quando esegui Recovery SVM tramite Trident.

- "-identity-preserve true" replica l'intera configurazione SVM.

- "-discard-configs network" esclude LIF e impostazioni di rete correlate.
- "-identity-preserve false" replica solo i volumi e la configurazione di sicurezza.

Recovery SVM tramite Trident

Trident non rileva automaticamente i guasti della SVM. In caso di disastro, l'amministratore può avviare manualmente il failover di Trident sulla nuova SVM.

Passaggi

1. Annulla i trasferimenti SnapMirror pianificati e in corso, interrompi la relazione di replicazione, arresta l'SVM di origine e poi attiva l'SVM di destinazione SnapMirror.
2. Se hai specificato `-identity-preserve false` o `-discard-config network` durante la configurazione della replica SVM, aggiorna `managementLIF` e `dataLIF` nel file di definizione del backend Trident.
3. Conferma `storagePrefix` è presente nel file di definizione del backend Trident. Questo parametro non può essere modificato. Omettere `storagePrefix` causerà il fallimento dell'aggiornamento del backend.
4. Aggiorna tutti i backend richiesti per riflettere il nuovo nome SVM di destinazione utilizzando:

```
./tridentctl update backend <backend-name> -f <backend-json-file> -n
<namespace>
```

5. Se hai specificato `-identity-preserve false` o `discard-config network`, devi rimbalzare tutti i pod dell'applicazione.

> Se hai specificato `-identity-preserve true`, tutti i volumi forniti da Trident iniziano a servire dati quando la SVM di destinazione viene attivata.

Replicazione e recovery del volume

Trident non può configurare le relazioni di replica di SnapMirror, tuttavia l'amministratore dello storage può utilizzare "Replica e recovery ONTAP SnapMirror" per replicare i volumi creati da Trident.

È quindi possibile importare i volumi recuperati in Trident utilizzando "tridentctl volume import".

> L'importazione non è supportata sui `ontap-nas-economy`, `ontap-san-economy` o `ontap-flexgroup-economy` driver.

Protezione dei dati Snapshot

È possibile proteggere e ripristinare i dati utilizzando:

- Un controller di snapshot esterno e CRD per creare snapshot di volumi Kubernetes di Persistent Volumes (PV).

  "Istantanee del volume"

- ONTAP Snapshot per ripristinare l'intero contenuto di un volume o per recuperare singoli file o LUN.

"ONTAP Snapshot"

Automatizzare il failover delle applicazioni stateful con Trident

La funzionalità di force-detach di Trident consente di staccare automaticamente i volumi dai nodi non integri in un cluster Kubernetes, prevenendo la corruzione dei dati e garantendo la disponibilità delle applicazioni. Questa funzionalità è particolarmente utile negli scenari in cui i nodi diventano non responsivi o vengono messi offline per manutenzione.

Dettagli sul force detach

La disconnessione forzata è disponibile per `ontap-san`, `ontap-san-economy`, `ontap-nas` e `ontap-nas-economy` solo. Prima di abilitare la disconnessione forzata, è necessario abilitare l'arresto non regolare dei nodi (NGNS) sul cluster Kubernetes. NGNS è abilitato per impostazione predefinita per Kubernetes 1.28 e versioni successive. Per ulteriori informazioni, fare riferimento a "Kubernetes: arresto non corretto del nodo".

Quando si utilizza il `ontap-nas` o `ontap-nas-economy` driver, è necessario impostare il parametro `autoExportPolicy` nella configurazione del backend su `true` in modo che Trident possa limitare l'accesso dal nodo Kubernetes con il taint applicato utilizzando policy di esportazione gestite.

Poiché Trident si basa su Kubernetes NGNS, non rimuovere `out-of-service` le taint da un nodo non integro finché tutti i carichi di lavoro non tollerabili non vengono riprogrammati. L'applicazione o la rimozione sconsiderata delle taint può compromettere la protezione dei dati backend.

Quando l'amministratore del cluster Kubernetes ha applicato la `node.kubernetes.io/out-of-service=nodeshutdown:NoExecute` taint al nodo e `enableForceDetach` è impostato su `true`, Trident determinerà lo stato del nodo e:

1. Interrompi l'accesso I/O backend per i volumi montati su quel nodo.
2. Contrassegna l'oggetto nodo Trident come `dirty` (non sicuro per nuove pubblicazioni).

Il controller Trident rifiuterà le nuove richieste di pubblicazione di volumi finché il nodo non verrà riqualificato (dopo essere stato contrassegnato come `dirty`) dal pod Trident del nodo. Qualsiasi carico di lavoro pianificato con un PVC montato (anche dopo che il nodo del cluster è integro e pronto) non verrà accettato finché Trident non potrà verificare il nodo `clean` (sicuro per nuove pubblicazioni).

Quando la salute del nodo viene ripristinata e la contaminazione viene rimossa, Trident:

1. Identificare e pulire i percorsi pubblicati obsoleti sul nodo.
2. Se il nodo si trova in uno stato `cleanable` (la contaminazione fuori servizio è stata rimossa e il nodo è in stato `Ready`) e tutti i percorsi pubblicati obsoleti sono puliti, Trident riammetterà il nodo come `clean` e consentirà nuovi volumi pubblicati sul nodo.

Dettagli sul failover automatico

È possibile automatizzare il processo di distacco forzato tramite integrazione con "operatore node health check (NHC)". Quando si verifica un errore di nodo, NHC attiva la Trident node remediation (TNR) e forza il distacco automaticamente creando una TridentNodeRemediation CR nello spazio dei nomi di Trident che definisce il nodo in errore. La TNR viene creata solo in caso di errore del nodo e rimossa da NHC una volta che il nodo torna online o viene eliminato.

Processo di rimozione del pod del nodo non riuscito

Il failover automatico seleziona i carichi di lavoro da rimuovere dal nodo in errore. Quando viene creato un TNR, il controller TNR contrassegna il nodo come dirty, impedendo qualsiasi nuova pubblicazione di volumi e inizia a rimuovere i pod supportati dal force-detach e i relativi allegati ai volumi.

Tutti i volumi/PVC supportati da force-detach sono supportati da automated-failover:

- Volumi NAS e NAS-economy che utilizzano policy di auto-export (SMB non è ancora supportato).
- SAN e volumi SAN-economy.

Fare riferimento a Dettagli sul force detach.

Comportamento predefinito:

- I pod che utilizzano volumi supportati dal force-detach vengono rimossi dal nodo in errore. Kubernetes ripianificherà questi pod su un nodo funzionante.
- I pod che utilizzano un volume non supportato da force-detach, inclusi i volumi non Trident, non vengono rimossi dal nodo in errore.
- I pod senza stato (non PVC) non vengono rimossi dal nodo non riuscito, a meno che l'annotazione del pod `trident.netapp.io/podRemediationPolicy: delete` sia impostata.

Sostituzione del comportamento di rimozione del pod:

Il comportamento di rimozione del pod può essere personalizzato utilizzando un'annotazione del pod: `trident.netapp.io/podRemediationPolicy[retain, delete]`. Queste annotazioni vengono esaminate e utilizzate quando si verifica un failover. Applica le annotazioni alla specifica del pod di deployment/replicaset di Kubernetes per evitare che l'annotazione scompaia dopo un failover:

- `retain` - Il pod NON verrà rimosso dal nodo in errore durante un failover automatico.
- `delete` - Il pod verrà rimosso dal nodo in errore durante un failover automatico.

Queste annotazioni possono essere applicate a qualsiasi pod.

- Le operazioni di I/O verranno bloccate solo sui nodi guasti per i volumi che supportano force-detach.
- Per i volumi che non supportano il force-detach, esiste il rischio di corruzione dei dati e problemi di multi-attach.

CR TridentNodeRemediation

Il CR TridentNodeRemediation (TNR) definisce un nodo guasto. Il nome del TNR è il nome del nodo guasto.

Esempio TNR:

```
apiVersion: trident.netapp.io/v1
kind: TridentNodeRemediation
metadata:
  name: <K8s-node-name>
spec: {}
```

TNR states: utilizzare i seguenti comandi per visualizzare lo stato dei TNR:
`kubectl get tnr <name> -n <trident-namespace>`

I TNR possono trovarsi in uno dei seguenti stati:

- *Rimediando*:
  - Interrompi l'accesso I/O backend per i volumi supportati da force-detach montati su quel nodo.
  - L'oggetto nodo Trident è contrassegnato come sporco (non sicuro per nuove pubblicazioni).
  - Rimuovi i pod e gli allegati di volume dal nodo
- *NodeRecoveryPending*:
  - Il controller attende che il nodo torni online.
  - Una volta che il nodo è online, publish-enforcement garantirà che il nodo sia pulito e pronto per nuove pubblicazioni di volumi.
- Se il nodo viene eliminato da K8s, il controller TNR rimuoverà il TNR e cesserà la riconciliazione.
- *Riuscito*:
  - Tutti i passaggi di remediation e ripristino del nodo sono stati completati con successo. Il nodo è pulito e pronto per nuove pubblicazioni di volumi.
- *Non riuscito*:
  - Errore irreversibile. I motivi dell'errore sono impostati nel campo status.message della CR.

Abilitazione del failover automatico

Prerequisiti:

- Assicurarsi che la disconnessione forzata sia abilitata prima di abilitare il failover automatico. Per ulteriori informazioni, fare riferimento a Dettagli sul force detach.
- Installare il controllo dello stato del nodo (NHC) nel cluster Kubernetes.
  - "Installa operator-sdk".
  - Installare Operator Lifecycle Manager (OLM) nel cluster se non è già installato: `operator-sdk olm install`.
  - Installa l'operatore Node Health check: `kubectl create -f https://operatorhub.io/install/node-healthcheck-operator.yaml`.

> È anche possibile utilizzare metodi alternativi per rilevare gli errori dei nodi, come specificato nella sezione [Integrating Custom Node Health Check Solutions] qui sotto.

Vedi "Operatore Node Health Check" per ulteriori informazioni.

Passaggi

1. Crea una CR NodeHealthCheck (NHC) nel namespace Trident per monitorare i nodi worker nel cluster. Esempio:

```
apiVersion: remediation.medik8s.io/v1alpha1
kind: NodeHealthCheck
metadata:
  name: <CR name>
spec:
  selector:
    matchExpressions:
      - key: node-role.kubernetes.io/control-plane
        operator: DoesNotExist
      - key: node-role.kubernetes.io/master
        operator: DoesNotExist
  remediationTemplate:
    apiVersion: trident.netapp.io/v1
    kind: TridentNodeRemediationTemplate
    namespace: <Trident installation namespace>
    name: trident-node-remediation-template
  minHealthy: 0 # Trigger force-detach upon one or more node failures
  unhealthyConditions:
    - type: Ready
      status: "False"
      duration: 0s
    - type: Ready
      status: Unknown
      duration: 0s
```

2. Applica il controllo di integrità del nodo CR nel namespace `trident`.

```
kubectl apply -f <nhc-cr-file>.yaml -n <trident-namespace>
```

Il CR sopra indicato è configurato per monitorare i nodi worker di K8s per le condizioni Ready: false e Unknown. Il failover automatico verrà attivato quando un nodo passa allo stato Ready: false o Ready: Unknown.

Il `unhealthyConditions` nella CR utilizza un grace period di 0 secondi. Ciò fa sì che il failover automatico venga attivato immediatamente quando K8s imposta la condizione del nodo Ready: false, che viene impostata dopo che K8s perde l'heartbeat da un nodo. K8s ha un valore predefinito di 40 secondi di attesa dopo l'ultimo heartbeat prima di impostare Ready: false. Questo grace period può essere personalizzato nelle opzioni di deployment di K8s.

Per ulteriori opzioni di configurazione, fare riferimento a "Documentazione di Node-Healthcheck-Operator".

Informazioni aggiuntive sulla configurazione

Quando Trident viene installato con force-detach abilitato, vengono create automaticamente due risorse aggiuntive nello spazio dei nomi Trident per facilitare l'integrazione con NHC: TridentNodeRemediationTemplate (TNRT) e ClusterRole.

TridentNodeRemediationTemplate (TNRT):

Il TNRT funge da modello per il controller NHC, che utilizza TNRT per generare risorse TNR secondo necessità.

```
apiVersion: trident.netapp.io/v1
kind: TridentNodeRemediationTemplate
metadata:
  name: trident-node-remediation-template
  namespace: trident
spec:
  template:
    spec: {}
```

ClusterRole:

Un ruolo cluster viene anche aggiunto durante l'installazione quando il force-detach è abilitato. Questo dà a NHC i permessi sui TNR nel namespace Trident.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  labels:
    rbac.ext-remediation/aggregate-to-ext-remediation: "true"
  name: tridentnoderemediation-access
rules:
- apiGroups:
  - trident.netapp.io
  resources:
  - tridentnoderemediationtemplates
  - tridentnoderemediations
  verbs:
  - get
  - list
  - watch
  - create
  - update
  - patch
  - delete
```

Aggiornamenti e manutenzione del cluster K8s

Per evitare eventuali failover, sospendere il failover automatico durante la manutenzione o gli aggiornamenti di K8s, quando è previsto che i nodi si fermino o si riavviino. È possibile sospendere il CR NHC (descritto sopra) applicando una patch al relativo CR:

```
kubectl patch NodeHealthCheck <cr-name> --patch
'{"spec":{"pauseRequests":["<description-for-reason-of-pause>"]}}' --type=merge
```

Questa operazione sospende il failover automatico. Per riattivare il failover automatico, rimuovere pauseRequests dalle specifiche dopo il completamento della manutenzione.

Limitazioni

- Le operazioni di I/O vengono impedite solo sui nodi non riusciti per i volumi supportati da force-detach. Solo i pod che utilizzano volumi/PVC supportati da force-detach vengono rimossi automaticamente.
- Automatic-failover e force-detach vengono eseguiti all'interno del pod trident-controller. Se il nodo che ospita trident-controller si guasta, l'automatic-failover sarà ritardato finché K8s non sposterà il pod su un nodo funzionante.

Integrazione di soluzioni personalizzate per il controllo dello stato dei nodi

È possibile sostituire Node Healthcheck Operator con strumenti alternativi di rilevamento dei guasti dei nodi per attivare il failover automatico. Per garantire la compatibilità con il meccanismo di failover automatizzato, la soluzione personalizzata dovrebbe:

- Crea un TNR quando viene rilevato un errore del nodo, utilizzando il nome del nodo non funzionante come nome CR del TNR.
- Eliminare il TNR quando il nodo è stato ripristinato e il TNR è nello stato Succeeded.

Sicurezza

Sicurezza

Utilizzare le raccomandazioni elencate qui per garantire che l'installazione di Trident sia sicura.

Esegui Trident nel suo namespace

È importante impedire alle applicazioni, agli amministratori delle applicazioni, agli utenti e alle applicazioni di gestione di accedere alle definizioni degli oggetti Trident o ai pod per garantire uno storage affidabile e bloccare potenziali attività dannose.

Per separare le altre applicazioni e gli altri utenti da Trident, installa sempre Trident nel proprio namespace Kubernetes (`trident`). Mettere Trident nel proprio namespace garantisce che solo il personale amministrativo di Kubernetes abbia accesso al pod Trident e agli artefatti (come backend e segreti CHAP, se applicabile) archiviati negli oggetti CRD con namespace. Devi assicurarti di consentire solo agli amministratori l'accesso al namespace Trident e quindi l'accesso all'applicazione `tridentctl`.

Utilizzare l'autenticazione CHAP con i backend ONTAP SAN

Trident supporta l'autenticazione basata su CHAP per i carichi di lavoro ONTAP SAN (utilizzando i `ontap-san`

e `ontap-san-economy` driver). NetApp consiglia di utilizzare CHAP bidirezionale con Trident per l'autenticazione tra un host e il backend di storage.

Per i backend ONTAP che utilizzano i driver di storage SAN, Trident può configurare il CHAP bidirezionale e gestire i nomi utente e i segreti CHAP tramite `tridentctl`. Fare riferimento a "Prepararsi a configurare il backend con i driver ONTAP SAN" per comprendere come Trident configura il CHAP sui backend ONTAP.

Utilizzare l'autenticazione CHAP con NetApp HCI e SolidFire backends

NetApp consiglia di implementare CHAP bidirezionale per garantire l'autenticazione tra un host e i backend NetApp HCI e SolidFire. Trident utilizza un oggetto segreto che include due password CHAP per tenant. Quando Trident è installato, gestisce i segreti CHAP e li memorizza in un oggetto CR `tridentvolume` per il rispettivo PV. Quando si crea un PV, Trident utilizza i segreti CHAP per avviare una sessione iSCSI e comunicare con il sistema NetApp HCI e SolidFire tramite CHAP.

> I volumi che vengono creati da Trident non sono associati ad alcun Volume Access Group.

Usa Trident con NVE e NAE

NetApp ONTAP fornisce la crittografia dei dati a riposo per proteggere i dati sensibili in caso di furto, restituzione o riutilizzo di un disco. Per dettagli, consultare "Panoramica sulla configurazione della crittografia del volume NetApp".

- Se NAE è abilitato sul backend, qualsiasi volume fornito in Trident sarà NAE-enabled.
  - È possibile impostare il flag di crittografia NVE su `""` per creare volumi abilitati per NAE.
- Se NAE non è abilitato sul backend, qualsiasi volume fornito in Trident sarà abilitato per NVE, a meno che il flag di crittografia NVE non sia impostato su `false` (il valore predefinito) nella configurazione del backend.

> I volumi creati in Trident su un backend abilitato NAE devono essere crittografati NVE o NAE.
>
> - È possibile impostare il flag di crittografia NVE su `true` nella configurazione del backend Trident per ignorare la crittografia NAE e utilizzare una chiave di crittografia specifica per ogni volume.
> - Impostando il flag di crittografia NVE `false` su un backend abilitato per NAE si crea un volume abilitato per NAE. Non è possibile disabilitare la crittografia NAE impostando il flag di crittografia NVE a `false`.

- È possibile creare manualmente un volume NVE in Trident impostando esplicitamente il flag di crittografia NVE su `true`.

Per maggiori informazioni sulle opzioni di configurazione del backend, fare riferimento a:

- "Opzioni di configurazione SAN ONTAP"
- "Opzioni di configurazione NAS ONTAP"

Linux Unified Key Setup (LUKS)

È possibile abilitare Linux Unified Key Setup (LUKS) per crittografare i volumi ONTAP SAN e ONTAP SAN ECONOMY su Trident. Trident supporta la rotazione della

passphrase e l'espansione del volume per i volumi crittografati con LUKS.

In Trident, i volumi crittografati con LUKS utilizzano il cifrario e la modalità aes-xts-plain64, come raccomandato da "NIST".

> La crittografia LUKS non è supportata per i sistemi ASA r2. Per informazioni sui sistemi ASA r2, vedere "Scopri i sistemi di storage ASA r2".

Prima di iniziare

- I nodi worker devono avere installato cryptsetup 2.1 o superiore (ma inferiore a 3.0). Per ulteriori informazioni, visitare "Gitlab: cryptsetup".
- Per motivi di prestazioni, NetApp raccomanda che i nodi worker supportino Advanced Encryption Standard New Instructions (AES-NI). Per verificare il supporto di AES-NI, eseguire il seguente comando:

```
grep "aes" /proc/cpuinfo
```

Se non viene restituito nulla, il processore non supporta AES-NI. Per ulteriori informazioni su AES-NI, visitare: "Intel: Advanced Encryption Standard Instructions (AES-NI)".

Abilita la crittografia LUKS

È possibile abilitare la crittografia per volume, lato host, utilizzando Linux Unified Key Setup (LUKS) per ONTAP SAN e ONTAP SAN ECONOMY volumi.

Passaggi

1. Definire gli attributi di crittografia LUKS nella configurazione backend. Per ulteriori informazioni sulle opzioni di configurazione backend per ONTAP SAN, fare riferimento a "Opzioni di configurazione SAN ONTAP".

```
{
  "storage": [
    {
      "labels": {
        "luks": "true"
      },
      "zone": "us_east_1a",
      "defaults": {
        "luksEncryption": "true"
      }
    },
    {
      "labels": {
        "luks": "false"
      },
      "zone": "us_east_1a",
      "defaults": {
        "luksEncryption": "false"
      }
    }
  ]
}
```

2. Utilizzare `parameters.selector` per definire i pool di archiviazione utilizzando la crittografia LUKS. Ad esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: csi.trident.netapp.io
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

3. Crea un segreto che contenga la passphrase LUKS. Ad esempio:

```
kubectl -n trident create -f luks-pvc1.yaml
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: A
  luks-passphrase: secretA
```

Limitazioni

I volumi crittografati LUKS non possono sfruttare la deduplicazione e la compressione di ONTAP.

Configurazione backend per l'importazione di volumi LUKS

Per importare un volume LUKS, è necessario impostare `luksEncryption` su (`true` sul backend. L'opzione `luksEncryption` indica a Trident se il volume è LUKS-compliant (`true` o non LUKS-compliant (`false`, come mostrato nell'esempio seguente.

```
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: trident_svm
username: admin
password: password
defaults:
  luksEncryption: 'true'
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

Configurazione PVC per l'importazione di volumi LUKS

Per importare i volumi LUKS in modo dinamico, impostare l'annotazione `trident.netapp.io/luksEncryption` a `true` e includere una storage class abilitata per LUKS nel PVC come mostrato in questo esempio.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: luks-pvc
  namespace: trident
  annotations:
    trident.netapp.io/luksEncryption: "true"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: luks-sc
```

Ruota una passphrase LUKS

È possibile ruotare la passphrase LUKS e confermare la rotazione.

> Non dimenticare una passphrase finché non hai verificato che non sia più referenziata da alcun volume, snapshot o segreto. Se una passphrase referenziata viene persa, potresti non essere in grado di montare il volume e i dati rimarranno crittografati e inaccessibili.

Informazioni su questa attività

La rotazione della passphrase LUKS avviene quando viene creato un pod che monta il volume dopo che è stata specificata una nuova passphrase LUKS. Quando viene creato un nuovo pod, Trident confronta la passphrase LUKS sul volume con la passphrase attiva nel secret.

- Se la passphrase sul volume non corrisponde alla passphrase attiva nel secret, si verifica la rotazione.
- Se la passphrase sul volume corrisponde alla passphrase attiva nel segreto, il `previous-luks-passphrase` parametro viene ignorato.

Passaggi

1. Aggiungere i `node-publish-secret-name` e `node-publish-secret-namespace` parametri StorageClass. Ad esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-san
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/backendType: "ontap-san"
  csi.storage.k8s.io/node-stage-secret-name: luks
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-publish-secret-name: luks
  csi.storage.k8s.io/node-publish-secret-namespace: ${pvc.namespace}
```

2. Identificare le passphrase esistenti sul volume o sullo snapshot.

Volume

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["A"]
```

Snapshot

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames:["A"]
```

3. Aggiornare il segreto LUKS per il volume per specificare la nuova e la precedente passphrase. Assicurarsi che `previous-luke-passphrase-name` e `previous-luks-passphrase` corrispondano alla passphrase precedente.

```
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: B
  luks-passphrase: secretB
  previous-luks-passphrase-name: A
  previous-luks-passphrase: secretA
```

4. Crea un nuovo pod che monta il volume. Questo è necessario per avviare la rotazione.
5. Verificare che la passphrase sia stata ruotata.

Volume

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["B"]
```

Snapshot

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames:["B"]
```

Risultati

La passphrase è stata ruotata quando sul volume e sullo snapshot viene restituita solo la nuova passphrase.

> Se vengono restituite due passphrase, ad esempio `luksPassphraseNames: ["B", "A"]`, la rotazione è incompleta. Puoi attivare un nuovo pod per tentare di completare la rotazione.

Abilita l'espansione del volume

È possibile abilitare l'espansione del volume su un volume crittografato con LUKS.

Passaggi

1. Abilita la funzionalità `CSINodeExpandSecret` feature gate (beta 1.25+). Consulta "Kubernetes 1.25: utilizzare i Secrets per l'espansione dei volumi CSI basata sui nodi" per i dettagli.
2. Aggiungere i `node-expand-secret-name` e `node-expand-secret-namespace` parametri StorageClass. Ad esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: csi.trident.netapp.io
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-expand-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-expand-secret-namespace: ${pvc.namespace}
allowVolumeExpansion: true
```

Risultati

Quando si avvia l'espansione dello storage online, il kubelet passa le credenziali appropriate al driver.

Crittografia in volo Kerberos

Utilizzando la crittografia Kerberos in volo, puoi migliorare la sicurezza dell'accesso ai dati abilitando la crittografia per il traffico tra il tuo cluster gestito e il backend di storage.

Trident supporta la crittografia Kerberos per ONTAP come storage backend:

- **On-premise ONTAP** - Trident supporta la crittografia Kerberos su connessioni NFSv3 e NFSv4 da cluster Red Hat OpenShift e Kubernetes upstream a volumi ONTAP on-premise.

È possibile creare, eliminare, ridimensionare, creare snapshot, clonare, clonare in sola lettura e importare volumi che utilizzano NFS con crittografia.

Configura la crittografia Kerberos in volo con i volumi ONTAP on-premise

È possibile abilitare la crittografia Kerberos sul traffico di storage tra il cluster gestito e un backend di storage ONTAP on-premise.

> La crittografia Kerberos per il traffico NFS con backend di storage on-premise ONTAP è supportata solo utilizzando il `ontap-nas` storage driver.

Prima di iniziare

- Assicurati di avere accesso all' `tridentctl` utility.
- Assicurati di avere accesso come amministratore al backend di storage ONTAP.
- Assicurati di conoscere il nome del volume o dei volumi che condividerai dal backend di storage ONTAP.
- Assicurarsi di aver preparato la macchina virtuale di storage ONTAP per supportare la crittografia Kerberos per i volumi NFS. Consultare "Abilitare Kerberos su un dataLIF" per le istruzioni.
- Assicurarsi che i volumi NFSv4 utilizzati con la crittografia Kerberos siano configurati correttamente. Fare riferimento alla sezione NetApp NFSv4 Domain Configuration (pagina 13) di "NetApp Miglioramenti NFSv4 e Guida alle Best Practice".

Aggiungi o modifica le policy di esportazione ONTAP

È necessario aggiungere regole alle policy di esportazione ONTAP esistenti o creare nuove policy di esportazione che supportino la crittografia Kerberos per il volume root della storage VM ONTAP, così come per qualsiasi volume ONTAP condiviso con il cluster Kubernetes upstream. Le regole delle policy di esportazione che aggiungi, o le nuove policy di esportazione che crei, devono supportare i seguenti protocolli di accesso e permessi di accesso:

Protocolli di accesso

Configura la policy di esportazione con i protocolli di accesso NFS, NFSv3 e NFSv4.

Dettagli di accesso

È possibile configurare una delle tre diverse versioni della crittografia Kerberos, a seconda delle esigenze per il volume:

- **Kerberos 5** - (autenticazione e crittografia)
- **Kerberos 5i** - (autenticazione e crittografia con protezione dell'identità)
- **Kerberos 5p** - (autenticazione e crittografia con protezione dell'identità e della privacy)

Configurare la regola della policy di esportazione ONTAP con le autorizzazioni di accesso appropriate. Ad esempio, se i cluster monteranno i volumi NFS con un misto di crittografia Kerberos 5i e Kerberos 5p, utilizzare le seguenti impostazioni di accesso:

Tipo	Accesso in sola lettura	Accesso in lettura/scrittura	Accesso come superutente
UNIX	Abilitato	Abilitato	Abilitato
Kerberos 5i	Abilitato	Abilitato	Abilitato
Kerberos 5p	Abilitato	Abilitato	Abilitato

Consultare la seguente documentazione per informazioni su come creare le policy di esportazione ONTAP e le regole delle policy di esportazione:

- "Creare una policy di esportazione"
- "Aggiungere una regola a una policy di esportazione"

Creare un backend di storage

È possibile creare una configurazione del backend di storage Trident che includa la funzionalità di crittografia Kerberos.

Informazioni su questa attività

Quando si crea un file di configurazione del backend di archiviazione che configura la crittografia Kerberos, è possibile specificare una delle tre diverse versioni della crittografia Kerberos utilizzando il parametro `spec.nfsMountOptions`:

- `spec.nfsMountOptions: sec=krb5` (autenticazione e crittografia)
- `spec.nfsMountOptions: sec=krb5i` (autenticazione e crittografia con protezione dell'identità)
- `spec.nfsMountOptions: sec=krb5p` (autenticazione e crittografia con protezione dell'identità e della privacy)

Specificare un solo livello Kerberos. Se si specifica più di un livello di crittografia Kerberos nell'elenco dei parametri, viene utilizzata solo la prima opzione.

Passaggi

1. Sul cluster gestito, creare un file di configurazione del backend di storage utilizzando il seguente esempio. Sostituire i valori tra parentesi <> con le informazioni del proprio ambiente:

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-ontap-nas-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  storageDriverName: "ontap-nas"
  managementLIF: <STORAGE_VM_MGMT_LIF_IP_ADDRESS>
  dataLIF: <PROTOCOL_LIF_FQDN_OR_IP_ADDRESS>
  svm: <STORAGE_VM_NAME>
  username: <STORAGE_VM_USERNAME_CREDENTIAL>
  password: <STORAGE_VM_PASSWORD_CREDENTIAL>
  nasType: nfs
  nfsMountOptions: ["sec=krb5i"] #can be krb5, krb5i, or krb5p
  qtreesPerFlexvol:
  credentials:
    name: backend-ontap-nas-secret
```

2. Utilizzare il file di configurazione creato nel passo precedente per creare il backend:

```
tridentctl create backend -f <backend-configuration-file>
```

Se la creazione del backend fallisce, c'è qualcosa di sbagliato nella configurazione del backend. Puoi visualizzare i log per determinare la causa eseguendo il seguente comando:

```
tridentctl logs
```

Dopo aver identificato e corretto il problema con il file di configurazione, è possibile eseguire nuovamente il comando create.

Creare una storage class

È possibile creare una storage class per il provisioning dei volumi con crittografia Kerberos.

Informazioni su questa attività

Quando si crea un oggetto classe di archiviazione, è possibile specificare una delle tre diverse versioni della crittografia Kerberos utilizzando il parametro `mountOptions`:

- `mountOptions: sec=krb5` (autenticazione e crittografia)
- `mountOptions: sec=krb5i` (autenticazione e crittografia con protezione dell'identità)
- `mountOptions: sec=krb5p` (autenticazione e crittografia con protezione dell'identità e della privacy)

Specificare un solo livello Kerberos. Se si specifica più di un livello di crittografia Kerberos nell'elenco dei parametri, viene utilizzata solo la prima opzione. Se il livello di crittografia specificato nella configurazione del backend di storage è diverso dal livello specificato nell'oggetto storage class, l'oggetto storage class ha la precedenza.

Passaggi

1. Crea un oggetto StorageClass Kubernetes, utilizzando il seguente esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-sc
provisioner: csi.trident.netapp.io
mountOptions:
  - sec=krb5i #can be krb5, krb5i, or krb5p
parameters:
  backendType: ontap-nas
  storagePools: ontapnas_pool
  trident.netapp.io/nasType: nfs
allowVolumeExpansion: true
```

2. Crea la classe di archiviazione:

```
kubectl create -f sample-input/storage-class-ontap-nas-sc.yaml
```

3. Assicurati che la storage class sia stata creata:

```
kubectl get sc ontap-nas-sc
```

   Dovresti vedere un output simile al seguente:

```
NAME          PROVISIONER             AGE
ontap-nas-sc    csi.trident.netapp.io   15h
```

Provisioning dei volumi

Dopo aver creato un backend di archiviazione e una classe di archiviazione, è possibile eseguire il provisioning di un volume. Per istruzioni, consultare "Effettua il provisioning di un volume".

Configura la crittografia Kerberos in volo con i volumi Azure NetApp Files

È possibile abilitare la crittografia Kerberos sul traffico di archiviazione tra il cluster gestito e un singolo backend di archiviazione Azure NetApp Files o un pool virtuale di backend di archiviazione Azure NetApp Files.

Prima di iniziare

- Assicurarsi di aver abilitato Trident sul cluster gestito Red Hat OpenShift.
- Assicurati di avere accesso all' `tridentctl` utility.
- Assicurati di aver preparato il backend di archiviazione Azure NetApp Files per la crittografia Kerberos, prendendo nota dei requisiti e seguendo le istruzioni in "Documentazione di Azure NetApp Files".
- Assicurarsi che i volumi NFSv4 utilizzati con la crittografia Kerberos siano configurati correttamente. Fare riferimento alla sezione NetApp NFSv4 Domain Configuration (pagina 13) di "NetApp Miglioramenti NFSv4 e Guida alle Best Practice".

Creare un backend di storage

È possibile creare una configurazione del backend di storage di Azure NetApp Files che include la funzionalità di crittografia Kerberos.

Informazioni su questa attività

Quando si crea un file di configurazione del backend di archiviazione che configura la crittografia Kerberos, è possibile definirlo in modo che venga applicato a uno dei due livelli possibili:

- Il **livello di backend dello storage** utilizzando il `spec.kerberos` campo
- Il **livello del pool virtuale** utilizzando il campo `spec.storage.kerberos`

Quando si definisce la configurazione a livello di pool virtuale, il pool viene selezionato utilizzando l'etichetta nella storage class.

A entrambi i livelli, puoi specificare una delle tre diverse versioni della crittografia Kerberos:

- `kerberos: sec=krb5` (autenticazione e crittografia)
- `kerberos: sec=krb5i` (autenticazione e crittografia con protezione dell'identità)
- `kerberos: sec=krb5p` (autenticazione e crittografia con protezione dell'identità e della privacy)

Passaggi

1. Sul cluster gestito, creare un file di configurazione del backend di archiviazione utilizzando uno dei seguenti esempi, a seconda di dove è necessario definire il backend di archiviazione (livello di backend di archiviazione o livello di pool virtuale). Sostituire i valori tra parentesi <> con le informazioni del proprio ambiente:

Esempio a livello di backend di storage

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-secret
```

Esempio di livello di pool virtuale

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  storage:
    - labels:
        type: encryption
      kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-secret
```

2. Utilizzare il file di configurazione creato nel passo precedente per creare il backend:

```
tridentctl create backend -f <backend-configuration-file>
```

Se la creazione del backend fallisce, c'è qualcosa di sbagliato nella configurazione del backend. Puoi visualizzare i log per determinare la causa eseguendo il seguente comando:

```
tridentctl logs
```

Dopo aver identificato e corretto il problema con il file di configurazione, è possibile eseguire nuovamente il comando create.

Creare una storage class

È possibile creare una storage class per il provisioning dei volumi con crittografia Kerberos.

Passaggi

1. Crea un oggetto StorageClass Kubernetes, utilizzando il seguente esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: azure-netapp-files
  trident.netapp.io/nasType: nfs
  selector: type=encryption
```

2. Crea la classe di archiviazione:

```
kubectl create -f sample-input/storage-class-sc-nfs.yaml
```

3. Assicurati che la storage class sia stata creata:

```
kubectl get sc -sc-nfs
```

Dovresti vedere un output simile al seguente:

```
NAME          PROVISIONER             AGE
sc-nfs        csi.trident.netapp.io   15h
```

Provisioning dei volumi

Dopo aver creato un backend di archiviazione e una classe di archiviazione, è possibile eseguire il provisioning di un volume. Per istruzioni, consultare "Effettua il provisioning di un volume".

Informazioni sul copyright

Informazioni sul marchio commerciale