Melhores práticas e recomendações

Trident

NetApp
July 01, 2026

Índice

Melhores práticas e recomendações

Implantação

Use as recomendações listadas aqui ao implantar Trident.

Implantar em um namespace dedicado

"Espaços de nomes" fornecem separação administrativa entre diferentes aplicações e são uma barreira para o compartilhamento de recursos. Por exemplo, um PVC de um namespace não pode ser consumido por outro. Trident fornece recursos de PV para todos os namespaces no cluster Kubernetes e, consequentemente, utiliza uma conta de serviço que possui privilégios elevados.

Além disso, o acesso ao Trident pod pode permitir que um usuário acesse as credenciais do sistema de storage e outras informações confidenciais. É importante garantir que os usuários do aplicativo e os aplicativos de gerenciamento não tenham a capacidade de acessar as definições de objeto do Trident ou os próprios pods.

Utilize quotas e limites de intervalo para controlar o consumo de armazenamento

O Kubernetes possui dois recursos que, quando combinados, fornecem um mecanismo poderoso para limitar o consumo de recursos por aplicativos. O "mecanismo de cota de armazenamento" permite que o administrador implemente limites globais e específicos de classe de armazenamento para capacidade e contagem de objetos por namespace. Além disso, usar um "limite de intervalo" garante que as solicitações de PVC estejam dentro de um valor mínimo e máximo antes de serem encaminhadas ao provisionador.

Esses valores são definidos para cada namespace individualmente, o que significa que cada namespace deve ter valores definidos que estejam de acordo com seus requisitos de recursos. Veja aqui para informações sobre "como aproveitar quotas".

Configuração de storage

Cada plataforma de storage no portfólio da NetApp possui recursos exclusivos que beneficiam aplicações, conteinerizadas ou não.

Visão geral da plataforma

Trident funciona com ONTAP e Element. Não existe uma plataforma que seja mais adequada para todas as aplicações e cenários do que outra, no entanto, as necessidades da aplicação e da equipe que administra o dispositivo devem ser levadas em consideração ao escolher uma plataforma.

Você deve seguir as melhores práticas básicas do sistema operacional com o protocolo que está utilizando. Opcionalmente, você pode considerar incorporar as melhores práticas da aplicação, quando disponíveis, com configurações de backend, storage class e PVC para otimizar o storage para aplicações específicas.

ONTAP e Cloud Volumes ONTAP melhores práticas

Aprenda as melhores práticas para configurar ONTAP e Cloud Volumes ONTAP para Trident.

As recomendações a seguir são diretrizes para configurar ONTAP para cargas de trabalho conteinerizadas, que consomem volumes provisionados dinamicamente pelo Trident. Cada uma deve ser considerada e

avaliada quanto à sua adequação ao seu ambiente.

Use SVM(s) dedicada(s) ao Trident

As Storage Virtual Machines (SVMs) fornecem isolamento e separação administrativa entre os tenants em um sistema ONTAP. Dedicar uma SVM a aplicativos permite a delegação de privilégios e possibilita aplicar as melhores práticas para limitar o consumo de recursos.

Existem diversas opções disponíveis para o gerenciamento da SVM:

- Forneça a interface de gerenciamento do cluster na configuração do backend, juntamente com as credenciais apropriadas, e especifique o nome da SVM.
- Crie uma interface de gerenciamento dedicada para a SVM usando ONTAP System Manager ou a CLI.
- Compartilhe a função de gerenciamento com uma interface de dados NFS.

Em cada caso, a interface deve estar no DNS e o nome DNS deve ser usado ao configurar Trident. Isso ajuda a facilitar alguns cenários de DR, por exemplo, SVM-DR sem o uso de retenção de identidade de rede.

Não há preferência entre ter uma LIF de gerenciamento dedicada ou compartilhada para a SVM, no entanto, você deve garantir que suas políticas de segurança de rede estejam alinhadas com a abordagem que você escolher. Independentemente disso, a LIF de gerenciamento deve ser acessível via DNS para facilitar a máxima flexibilidade caso "SVM-DR" seja usada em conjunto com Trident.

Limite a contagem máxima de volume

Os sistemas de storage ONTAP têm uma contagem de volume máxima, que varia de acordo com a versão do software e a plataforma de hardware. Consulte "NetApp Hardware Universe" para sua plataforma específica e versão do ONTAP para determinar os limites exatos. Quando a contagem de volume é atingida, as operações de provisionamento falham não apenas para Trident, mas para todas as solicitações de storage.

Os drivers do Trident `ontap-nas` e `ontap-san` provisionam um FlexVolume para cada Persistent Volume (PV) do Kubernetes criado. O driver `ontap-nas-economy` cria aproximadamente um FlexVolume para cada 200 PVs (configurável entre 50 e 300). O driver `ontap-san-economy` cria aproximadamente um FlexVolume para cada 100 PVs (configurável entre 50 e 200). Para evitar que o Trident consuma todos os volumes disponíveis no sistema de storage, você deve definir um limite no SVM. Você pode fazer isso pela linha de comando:

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

O valor para `max-volumes` varia de acordo com diversos critérios específicos do seu ambiente:

- O número de volumes existentes no cluster ONTAP
- O número de volumes que você espera provisionar fora do Trident para outros aplicativos
- O número de volumes persistentes que se espera que sejam consumidos por aplicações Kubernetes

O `max-volumes` valor é o total de volumes provisionados em todos os nós do cluster ONTAP, e não em um nó ONTAP individual. Como resultado, você pode encontrar algumas situações em que um nó de cluster ONTAP pode ter muito mais ou muito menos volumes provisionados pelo Trident do que outro nó.

Por exemplo, um cluster ONTAP de dois nós tem a capacidade de hospedar um máximo de 2000 FlexVol volumes. Ter a contagem máxima de volume definida para 1250 parece muito razoável. No entanto, se

apenas "agregados" de um nó forem atribuídos à SVM, ou se os agregados atribuídos de um nó não puderem ser provisionados (por exemplo, devido à capacidade), então o outro nó se torna o destino para todos os volumes provisionados pelo Trident. Isso significa que o limite de volume pode ser atingido para esse nó antes que o valor `max-volumes` seja alcançado, resultando em impacto tanto para o Trident quanto para outras operações de volume que utilizam esse nó. **Você pode evitar essa situação garantindo que os agregados de cada nó do cluster sejam atribuídos à SVM usada pelo Trident em números iguais.**

Clonar um volume

NetApp Trident suporta clonagem de volumes ao usar os `ontap-nas`, `ontap-san` e `solidfire-san` drivers de armazenamento. Ao usar os `ontap-nas-flexgroup` ou `ontap-nas-economy` drivers, a clonagem não é suportada. Criar um novo volume a partir de um volume existente resultará na criação de um novo snapshot.

Evite clonar um PVC associado a um StorageClass diferente. Realize operações de clonagem dentro do mesmo StorageClass para garantir a compatibilidade e evitar comportamentos inesperados.

Limite o tamanho máximo dos volumes criados pelo Trident

Para configurar o tamanho máximo para volumes que podem ser criados pelo Trident, use o `limitVolumeSize` parâmetro na sua `backend.json` definição.

Além de controlar o tamanho do volume no array de storage, você também deve aproveitar as capacidades do Kubernetes.

Limite o tamanho máximo de FlexVols criados pelo Trident

Para configurar o tamanho máximo dos FlexVols usados como pools para os drivers ontap-san-economy e ontap-nas-economy, use o `limitVolumePoolSize` parâmetro na sua `backend.json` definição.

Configurar o Trident para usar CHAP bidirecional

Você pode especificar os nomes de usuário e senhas do iniciador e do destino CHAP na definição do seu backend e fazer com que Trident habilite o CHAP na SVM. Usando o `useCHAP` parâmetro na configuração do seu backend, Trident autentica as conexões iSCSI para backends ONTAP com CHAP.

Crie e use uma política de QoS para SVM

Ao utilizar uma política de qualidade do serviço (QoS) do ONTAP, aplicada à SVM, limita-se o número de IOPS consumíveis pelos volumes provisionados pelo Trident. Isso ajuda a "prevenir um agressor" evitar que um contêiner descontrolado afete cargas de trabalho fora da SVM do Trident.

Você pode criar uma política de QoS para a SVM em algumas etapas. Consulte a documentação da sua versão do ONTAP para as informações mais precisas. O exemplo abaixo cria uma política de QoS que limita o total de IOPS disponível para a 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>
```

Além disso, se a sua versão do ONTAP for compatível, você pode considerar o uso de um QoS mínimo para garantir uma quantidade de largura de banda para cargas de trabalho em contêineres. O QoS adaptável não é compatível com uma política de nível SVM.

O número de IOPS dedicados às cargas de trabalho em contêineres depende de muitos aspectos. Entre outras coisas, incluem:

- Outras cargas de trabalho que utilizam o array de storage. Caso existam outras cargas de trabalho, não relacionadas à implantação do Kubernetes, que utilizem os recursos de storage, deve-se tomar cuidado para garantir que essas cargas de trabalho não sejam acidentalmente afetadas negativamente.
- Cargas de trabalho esperadas em execução em contêineres. Se cargas de trabalho com altos requisitos de IOPS forem executadas em contêineres, uma política de QoS baixa resulta em uma experiência ruim.

É importante lembrar que uma política de QoS atribuída no nível da SVM resulta em todos os volumes provisionados para a SVM compartilharem o mesmo pool de IOPS. Se uma, ou um pequeno número, das aplicações conteinerizadas tiver uma alta necessidade de IOPS, ela pode se tornar um problema para as outras cargas de trabalho conteinerizadas. Se for esse o caso, talvez você queira considerar o uso de automação externa para atribuir políticas de QoS por volume.

> Você deve atribuir o grupo de políticas de qualidade do serviço (QoS) à SVM **somente** se a sua versão do ONTAP for anterior à 9.8.

Criar grupos de políticas de QoS para Trident

A qualidade do serviço (QoS) garante que o desempenho de cargas de trabalho críticas não seja prejudicado por cargas de trabalho concorrentes. Os grupos de políticas de QoS do ONTAP fornecem opções de QoS para volumes e permitem que os usuários definam o limite de taxa de transferência para uma ou mais cargas de trabalho. Para mais informações sobre QoS, consulte "Garantindo a taxa de transferência com qualidade do serviço". Você pode especificar grupos de políticas de QoS no backend ou em um pool de storage, e eles são aplicados a cada volume criado nesse pool ou backend.

ONTAP possui dois tipos de grupos de políticas de qualidade do serviço: tradicional e adaptável. Os grupos de políticas tradicionais fornecem uma taxa máxima (ou mínima, em versões posteriores) de IOPS fixa. A qualidade do serviço adaptável dimensiona automaticamente a taxa de transferência de acordo com o tamanho da carga de trabalho, mantendo a proporção de IOPS para TBs|GBs conforme o tamanho da carga de trabalho muda. Isso proporciona uma vantagem significativa ao gerenciar centenas ou milhares de cargas de trabalho em uma grande implementação.

Considere o seguinte ao criar grupos de políticas de QoS:

- Você deve definir a `qosPolicy` chave no bloco `defaults` da configuração do backend. Veja o seguinte exemplo de configuração do 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
```

- Você deve aplicar os grupos de políticas por volume, para que cada volume receba toda a taxa de transferência especificada pelo grupo de políticas. Grupos de políticas compartilhados não são suportados.

Para obter mais informações sobre grupos de políticas de qualidade do serviço (QoS), consulte "Referência de comandos ONTAP".

Limite o acesso aos recursos de storage aos membros do cluster Kubernetes

Limitar o acesso aos volumes NFS, LUNs iSCSI e LUNs FC criados pelo Trident é um componente crítico da postura de segurança da sua implementação do Kubernetes. Isso impede que hosts que não fazem parte do cluster Kubernetes acessem os volumes e potencialmente modifiquem dados inesperadamente.

É importante entender que os namespaces são o limite lógico para os recursos no Kubernetes. A premissa é que recursos no mesmo namespace podem ser compartilhados, no entanto, é importante ressaltar que não existe capacidade entre namespaces diferentes. Isso significa que, mesmo que os PVs sejam objetos globais, quando vinculados a um PVC, eles só podem ser acessados por pods que estejam no mesmo namespace. **É fundamental garantir que os namespaces sejam usados para fornecer separação quando apropriado.**

A principal preocupação da maioria das organizações em relação à segurança de dados em um contexto Kubernetes é que um processo em um contêiner possa acessar o armazenamento montado no host, mas que não se destina ao contêiner. "Espaços de nomes" são projetados para evitar esse tipo de comprometimento. No entanto, há uma exceção: contêineres privilegiados.

Um contêiner privilegiado é aquele que é executado com permissões de nível de host substancialmente maiores do que o normal. Essas permissões não são negadas por padrão, portanto, certifique-se de desativar a capacidade usando "políticas de segurança do pod".

Para volumes onde o acesso é desejado tanto do Kubernetes quanto de hosts externos, o armazenamento deve ser gerenciado de maneira tradicional, com o PV introduzido pelo administrador e não gerenciado pelo

Trident. Isso garante que o volume de armazenamento seja destruído somente quando tanto o Kubernetes quanto os hosts externos estiverem desconectados e não estiverem mais utilizando o volume. Além disso, uma política de exportação personalizada pode ser aplicada, permitindo o acesso a partir dos nós do cluster Kubernetes e de servidores direcionados fora do cluster Kubernetes.

Para implantações que possuem nós de infraestrutura dedicados (por exemplo, OpenShift) ou outros nós que não conseguem agendar aplicativos de usuário, políticas de exportação separadas devem ser usadas para limitar ainda mais o acesso aos recursos de storage. Isso inclui a criação de uma política de exportação para serviços que são implantados nesses nós de infraestrutura (por exemplo, os serviços de Métricas e Registro do OpenShift), e para aplicativos padrão que são implantados em nós que não fazem parte da infraestrutura.

Use uma política de exportação dedicada

Você deve garantir que exista uma política de exportação para cada backend que permita acesso apenas aos nós presentes no cluster Kubernetes. Trident pode criar e gerenciar políticas de exportação automaticamente. Dessa forma, Trident limita o acesso aos volumes que provisiona aos nós do cluster Kubernetes e simplifica a adição/remoção de nós.

Alternativamente, você também pode criar uma política de exportação manualmente e preenchê-la com uma ou mais regras de exportação que processem cada solicitação de acesso ao nó:

- Use o comando da CLI `vserver export-policy create` ONTAP para criar a política de exportação.
- Adicione regras à política de exportação usando o `vserver export-policy rule create` comando da CLI do ONTAP.

Executar esses comandos permite que você restrinja quais nós do Kubernetes têm acesso aos dados.

Desative `showmount` para o aplicativo SVM

O `showmount` recurso permite que um cliente NFS consulte a SVM para obter uma lista de exports NFS disponíveis. Um pod implantado no cluster Kubernetes pode executar o `showmount -e` comando contra a SVM e receber uma lista de montagens disponíveis, incluindo aquelas às quais não tem acesso. Embora isso, por si só, não represente uma falha de segurança, fornece informações desnecessárias que podem auxiliar um usuário não autorizado a se conectar a um export NFS.

Você deve desativar `showmount` usando o comando ONTAP CLI em nível de SVM:

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

Práticas recomendadas do SolidFire

Aprenda as melhores práticas para configurar o armazenamento SolidFire para Trident.

Criar conta SolidFire

Cada conta SolidFire representa um proprietário de volume único e recebe seu próprio conjunto de credenciais do Challenge-Handshake Authentication Protocol (CHAP). Você pode acessar volumes atribuídos a uma conta usando o nome da conta e as credenciais CHAP correspondentes ou por meio de um grupo de acesso a volumes. Uma conta pode ter até dois mil volumes atribuídos a ela, mas um volume pode pertencer a apenas uma conta.

Criar uma política de QoS

Utilize as políticas de qualidade do serviço (QoS) do SolidFire se você quiser criar e salvar uma configuração padronizada de qualidade do serviço que pode ser aplicada a vários volumes.

Você pode definir parâmetros de qualidade do serviço em uma base por volume. O desempenho de cada volume pode ser garantido definindo três parâmetros configuráveis que definem a qualidade do serviço: Min IOPS, Max IOPS e Burst IOPS.

Aqui estão os possíveis valores mínimos, máximos e de burst de IOPS para o tamanho de bloco de 4Kb.

Parâmetro IOPS	Definição	Valor mínimo	Valor padrão	Valor máximo (4Kb)
IOPS mínimo	O nível garantido de desempenho para um volume.	50	50	15000
IOPS máximos	O desempenho não ultrapassará esse limite.	50	15000	200.000
IOPS de rajada	Máximo de IOPS permitido em um cenário de pico curto.	50	15000	200.000

Embora os valores de Max IOPS e Burst IOPS possam ser configurados para até 200.000, o desempenho máximo real de um volume é limitado pelo uso do cluster e pelo desempenho por nó.

O tamanho do bloco e a largura de banda têm influência direta sobre o número de IOPS. À medida que os tamanhos dos blocos aumentam, o sistema aumenta a largura de banda para um nível necessário para processar os blocos maiores. À medida que a largura de banda aumenta, o número de IOPS que o sistema é capaz de atingir diminui. Consulte "Qualidade do serviço do SolidFire" para mais informações sobre QoS e desempenho.

SolidFire autenticação

O Element suporta dois métodos de autenticação: CHAP e Grupos de Acesso a Volumes (VAG). CHAP utiliza o protocolo CHAP para autenticar o host no backend. Grupos de Acesso a Volumes controla o acesso aos volumes que provisiona. NetApp recomenda o uso do CHAP para autenticação, pois é mais simples e não possui limites de escalabilidade.

Trident com o provisionador CSI aprimorado suporta o uso de autenticação CHAP. VAGs devem ser usados apenas no modo de operação tradicional não-CSI.

A autenticação CHAP (verificação de que o iniciador é o usuário pretendido do volume) é suportada apenas com controle de acesso baseado em conta. Se você estiver usando CHAP para autenticação, duas opções estão disponíveis: CHAP unidirecional e CHAP bidirecional. O CHAP unidirecional autentica o acesso ao volume usando o nome da conta SolidFire e o segredo do iniciador. A opção CHAP bidirecional oferece a maneira mais segura de autenticar o volume, pois o volume autentica o host por meio do nome da conta e do segredo do iniciador e, em seguida, o host autentica o volume por meio do nome da conta e do segredo de

destino.

No entanto, se o CHAP não puder ser habilitado e os VAGs forem necessários, crie o grupo de acesso e adicione os iniciadores de host e os volumes ao grupo de acesso. Cada IQN que você adicionar a um grupo de acesso pode acessar cada volume do grupo com ou sem autenticação CHAP. Se o iniciador iSCSI estiver configurado para usar autenticação CHAP, o controle de acesso baseado em conta será usado. Se o iniciador iSCSI não estiver configurado para usar autenticação CHAP, então o controle de acesso do Volume Access Group será usado.

Onde encontrar mais informações?

Algumas das melhores práticas de documentação estão listadas abaixo. Pesquise no "NetApp biblioteca" para as versões mais recentes.

ONTAP

- "NFS Guia de Melhores Práticas e Implementação"
- "Administração SAN" (para iSCSI)
- "Configuração iSCSI Express para RHEL"

Software Element

- "Configurando SolidFire para Linux"

NetApp HCI

- "NetApp HCI pré-requisitos de implantação"
- "Acesse o NetApp Deployment Engine"

Informações sobre as melhores práticas de aplicação

- "Melhores práticas para MySQL no ONTAP"
- "Melhores práticas para MySQL em SolidFire"
- "NetApp SolidFire e Cassandra"
- "Melhores práticas da Oracle sobre SolidFire"
- "Melhores práticas do PostgreSQL em SolidFire"

Nem todas as aplicações possuem diretrizes específicas, é importante trabalhar com sua NetApp equipe e usar o "NetApp biblioteca" para encontrar a documentação mais atualizada.

Integrar Trident

Para integrar Trident, os seguintes elementos de design e arquitetura precisam ser integrados: seleção e implantação de drivers, design de classes de armazenamento, design de pools virtuais, impactos do Persistent Volume Claim (PVC) no provisionamento de storage, operações de volume e implantação de serviços OpenShift usando Trident.

Seleção e implantação de driver

Selecione e implemente um driver de backend para seu sistema de storage.

Drivers de backend ONTAP

Os drivers de backend do ONTAP são diferenciados pelo protocolo utilizado e pela forma como os volumes são provisionados no sistema de storage. Portanto, considere cuidadosamente qual driver implantar.

Em um nível mais alto, se sua aplicação possui componentes que precisam de storage compartilhado (vários pods acessando o mesmo PVC), os drivers baseados em NAS seriam a escolha padrão, enquanto os drivers iSCSI baseados em bloco atendem às necessidades de storage não compartilhado. Escolha o protocolo com base nos requisitos da aplicação e no nível de familiaridade das equipes de storage e infraestrutura. De modo geral, há pouca diferença entre eles para a maioria das aplicações, então, frequentemente, a decisão se baseia na necessidade ou não de storage compartilhado (onde mais de um pod precisará de acesso simultâneo).

Os drivers de backend ONTAP disponíveis são:

- `ontap-nas`: Cada PV provisionado é um ONTAP FlexVolume.
- `ontap-nas-economy`: Cada PV provisionado é uma qtree, com um número configurável de qtrees por FlexVolume (o padrão é 200).
- `ontap-nas-flexgroup`: Cada PV provisionado como um ONTAP FlexGroup, e todos os agregados atribuídos a um SVM são utilizados.
- `ontap-san`: Cada PV provisionado é um LUN dentro de seu próprio FlexVolume.
- `ontap-san-economy`: Cada PV provisionado é um LUN, com um número configurável de LUNs por FlexVolume (o padrão é 100).

A escolha entre os três drivers NAS tem algumas ramificações nos recursos que são disponibilizados para o aplicativo.

Observe que, nas tabelas abaixo, nem todos os recursos são expostos pelo Trident. Alguns devem ser aplicados pelo administrador de storage após o provisionamento, caso essa funcionalidade seja desejada. As notas de rodapé em sobrescrito distinguem a funcionalidade por recurso e driver.

Drivers NAS do ONTAP	Instantâneos	Clones	Políticas de exportação dinâmicas	Multi-attach	QoS	Redimensionar	Replicação
`ontap-nas`	Sim	Sim	Sim nota de rodapé:5[]	Sim	Sim nota de rodapé:1[]	Sim	Sim nota de rodapé:1[]
`ontap-nas-economy`	NO [3]	NO [3]	Sim nota de rodapé:5[]	Sim	NO [3]	Sim	NO [3]
`ontap-nas-flexgroup`	Sim nota de rodapé:1[]	NÃO	Sim nota de rodapé:5[]	Sim	Sim nota de rodapé:1[]	Sim	Sim nota de rodapé:1[]

Trident oferece 2 drivers SAN para ONTAP, cujas capacidades são apresentadas abaixo.

Drivers SAN do ONTAP	Instantâneos	Clones	Multi-attach	CHAP bidirecional	QoS	Redimensionar	Replicação
`ontap-san`	Sim	Sim	Sim	Sim	Sim nota de rodapé:1[]	Sim	Sim nota de rodapé:1[]
`ontap-san-economy`	Sim	Sim	Sim	Sim	NO [3]	Sim	NO [3]

> Nota de rodapé para as tabelas acima: Sim [1]: Não gerenciado pelo Trident Sim [2]: Gerenciado pelo Trident, mas não granular em PV Nãonote:3[]: Não gerenciado pelo Trident e não granular em PV Sim [4]: Suportado para volumes raw-block Sim [5]: Suportado pelo Trident

As funcionalidades que não são granulares em relação ao PV são aplicadas a todo o FlexVolume e todos os PVs (ou seja, qtrees ou LUNs em FlexVols compartilhados) compartilharão um agendamento comum.

Como podemos ver nas tabelas acima, grande parte da funcionalidade entre o `ontap-nas` e o `ontap-nas-economy` é a mesma. No entanto, como o driver `ontap-nas-economy` limita a capacidade de controlar o agendamento na granularidade por PV, isso pode afetar especialmente o seu planejamento de recuperação de desastres e backup. Para equipes de desenvolvimento que desejam aproveitar a funcionalidade de clonagem de PVC no armazenamento ONTAP, isso só é possível ao usar os drivers `ontap-nas`, `ontap-san` ou `ontap-san-economy`.

O `solidfire-san`driver também é capaz de clonar PVCs.

Drivers de backend do Cloud Volumes ONTAP

Cloud Volumes ONTAP oferece controle de dados juntamente com recursos de armazenamento de nível empresarial para diversos casos de uso, incluindo compartilhamento de arquivos e armazenamento em nível de bloco, atendendo a protocolos NAS e SAN (NFS, SMB / CIFS e iSCSI). Os drivers compatíveis para Cloud Volume ONTAP são `ontap-nas`, `ontap-nas-economy`, `ontap-san` e `ontap-san-economy`. Estes são aplicáveis para Cloud Volume ONTAP para Azure, Cloud Volume ONTAP para GCP.

Drivers de backend do Amazon FSx for ONTAP

Amazon FSx for NetApp ONTAP permite que você aproveite os recursos, o desempenho e as capacidades administrativas do NetApp com os quais você já está familiarizado, enquanto aproveita a simplicidade, agilidade, segurança e escalabilidade de armazenar dados na AWS. FSx for ONTAP oferece suporte a muitos recursos do sistema de arquivos ONTAP e APIs de administração. Os drivers compatíveis para Cloud Volume ONTAP são `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `ontap-san` e `ontap-san-economy`.

NetApp HCI/SolidFire drivers de backend

O `solidfire-san` driver usado com as plataformas NetApp HCI/SolidFire ajuda o administrador a configurar um backend Element para Trident com base em limites de QoS. Se você deseja projetar seu backend para definir limites de QoS específicos nos volumes provisionados pelo Trident, use o parâmetro `type` no arquivo de backend. O administrador também pode restringir o tamanho do volume que pode ser criado no storage usando o parâmetro `limitVolumeSize`. Atualmente, recursos de storage do Element, como

redimensionamento e replicação de volumes, não são suportados pelo driver `solidfire-san`. Essas operações devem ser realizadas manualmente por meio da interface web do Element Software.

Driver SolidFire	Instantâneos	Clones	Multi-attach	CHAP	QoS	Redimensionar	Replicação
`solidfire-san`	Sim	Sim	Sim	Sim	Sim	Sim	Sim nota de rodapé:1[]

> Nota de rodapé: Simnota de rodapé:1[]: Não gerenciado pelo Trident Simnota de rodapé:2[]: Compatível com volumes de bloco bruto

Drivers de back-end do Azure NetApp Files

Trident usa o `azure-netapp-files` driver para gerenciar o serviço "Azure NetApp Files".

Mais informações sobre este driver e como configurar podem ser encontradas em "Configuração do backend Trident para Azure NetApp Files".

Driver do Azure NetApp Files	Instantâneos	Clones	Multi-attach	QoS	Expandir	Replicação
`azure-netapp-files`	Sim	Sim	Sim	Sim	Sim	Sim nota de rodapé:1[]

> Nota de rodapé: Sim [1]: Não gerenciado por Trident

Design de storage class

Classes de armazenamento individuais precisam ser configuradas e aplicadas para criar um objeto de classe de armazenamento do Kubernetes. Esta seção aborda como projetar uma classe de armazenamento para sua aplicação.

Utilização específica do backend

A filtragem pode ser usada dentro de um objeto de classe de armazenamento específico para determinar qual pool de storage ou conjunto de pools deve ser usado com essa classe de armazenamento específica. Três conjuntos de filtros podem ser definidos na Storage Class: `storagePools`, `additionalStoragePools`, e/ou `excludeStoragePools`.

O `storagePools` parâmetro ajuda a restringir o armazenamento ao conjunto de pools que correspondem a quaisquer atributos especificados. O `additionalStoragePools` parâmetro é usado para estender o conjunto de pools que Trident usa para provisionamento junto com o conjunto de pools selecionados pelos atributos e `storagePools` parâmetros. Você pode usar qualquer um dos parâmetros individualmente ou ambos juntos para garantir que o conjunto apropriado de pools de armazenamento seja selecionado.

O `excludeStoragePools`parâmetro é usado para excluir especificamente o conjunto listado de pools que correspondem aos atributos.

Emular políticas de QoS

Se você deseja projetar classes de armazenamento para emular políticas de Qualidade de Serviço, crie uma classe de armazenamento com o `media` atributo como `hdd` ou `ssd`. Com base no `media` atributo mencionado na storage class, Trident selecionará o backend apropriado que serve `hdd` ou `ssd` agregados para corresponder ao atributo de mídia e então direcionará o provisionamento dos volumes para o agregado específico. Portanto, podemos criar uma storage class PREMIUM que teria o `media` atributo definido como `ssd`, que poderia ser classificada como a política de QoS PREMIUM. Podemos criar outra storage class STANDARD que teria o atributo de mídia definido como `hdd`, que poderia ser classificada como a política de QoS STANDARD. Também podemos usar o atributo `IOPS` na storage class para redirecionar o provisionamento para um appliance Element, que pode ser definido como uma política de QoS.

Utilize backend com base em funcionalidades específicas

As classes de armazenamento podem ser projetadas para direcionar o provisionamento de volumes em um backend específico, onde recursos como provisionamento thin e thick, snapshots, clones e criptografia estão habilitados. Para especificar qual storage usar, crie Storage Classes que especifiquem o backend apropriado com o recurso necessário habilitado.

Pools virtuais

Pools virtuais estão disponíveis para todos os backends do Trident. Você pode definir pools virtuais para qualquer backend, usando qualquer driver que o Trident fornece.

Os pools virtuais permitem que um administrador crie um nível de abstração sobre backends que podem ser referenciados por meio de Storage Classes, para maior flexibilidade e alocação eficiente de volumes nos backends. Diferentes backends podem ser definidos com a mesma classe de serviço. Além disso, vários storage pools podem ser criados no mesmo backend, mas com características diferentes. Quando uma Storage Class é configurada com um seletor com rótulos específicos, Trident escolhe um backend que corresponda a todos os rótulos do seletor para alocar o volume. Se os rótulos do seletor da Storage Class corresponderem a vários storage pools, Trident escolherá um deles para provisionar o volume.

Design de pool virtual

Ao criar um backend, geralmente é possível especificar um conjunto de parâmetros. Era impossível para o administrador criar outro backend com as mesmas credenciais de storage e com um conjunto diferente de parâmetros. Com a introdução dos pools virtuais, esse problema foi atenuado. Um pool virtual é uma camada de abstração introduzida entre o backend e a Kubernetes Storage Class, permitindo que o administrador defina parâmetros juntamente com rótulos que podem ser referenciados por meio das Kubernetes Storage Classes como um seletor, de forma agnóstica ao backend. Pools virtuais podem ser definidos para todos os backends NetApp compatíveis com Trident. Essa lista inclui SolidFire/NetApp HCI, ONTAP, assim como Azure NetApp Files.

> Ao definir pools virtuais, recomenda-se não tentar reorganizar a ordem dos pools virtuais existentes em uma definição de backend. Também é aconselhável não editar/modificar atributos de um pool virtual existente e definir um novo pool virtual em vez disso.

Emulando diferentes níveis de serviço/QoS

É possível projetar pools virtuais para emular classes de serviço. Usando a implementação de pool virtual para Cloud Volume Service for Azure NetApp Files, vamos examinar como podemos configurar diferentes classes de serviço. Configure o backend do Azure NetApp Files com vários rótulos, representando diferentes níveis de desempenho. Defina o aspecto `servicelevel` para o nível de desempenho apropriado e adicione outros aspectos necessários sob cada rótulo. Agora crie diferentes Storage Classes do Kubernetes que serão

mapeadas para diferentes pools virtuais. Usando o campo `parameters.selector`, cada StorageClass especifica quais pools virtuais podem ser usados para hospedar um volume.

Atribuindo conjunto específico de aspectos

É possível projetar vários pools virtuais com um conjunto específico de aspectos a partir de um único backend de armazenamento. Para fazer isso, configure o backend com vários rótulos e defina os aspectos necessários em cada rótulo. Agora crie diferentes classes de armazenamento do Kubernetes usando o `parameters.selector` campo que mapeará para diferentes pools virtuais. Os volumes provisionados no backend terão os aspectos definidos no pool virtual escolhido.

Características do PVC que afetam o provisionamento de storage

Alguns parâmetros além da classe de armazenamento solicitada podem afetar o processo de decisão de provisionamento do Trident ao criar um PVC.

Modo de acesso

Ao solicitar armazenamento por meio de um PVC, um dos campos obrigatórios é o modo de acesso. O modo desejado pode afetar o backend selecionado para hospedar a solicitação de armazenamento.

Trident tentará encontrar uma correspondência entre o protocolo de storage utilizado e o método de acesso especificado, de acordo com a seguinte matriz. Isso é independente da plataforma de storage subjacente.

	ReadWriteOnce	ReadOnlyMany	ReadWriteMany
iSCSI	Sim	Sim	Sim (bloco bruto)
NFS	Sim	Sim	Sim

Uma solicitação de um ReadWriteMany PVC enviada a uma implantação do Trident sem um backend NFS configurado resultará em nenhum volume sendo provisionado. Por esse motivo, o solicitante deve usar o modo de acesso apropriado para sua aplicação.

Operações de volume

Modificar volumes persistentes

Os volumes persistentes são, com duas exceções, objetos imutáveis no Kubernetes. Uma vez criados, a política de recuperação e o tamanho podem ser modificados. No entanto, isso não impede que alguns aspectos do volume sejam modificados fora do Kubernetes. Isso pode ser desejável para personalizar o volume para aplicações específicas, garantir que a capacidade não seja consumida acidentalmente ou simplesmente mover o volume para um controlador de storage diferente por qualquer motivo.

> Os provisionadores integrados do Kubernetes não suportam operações de redimensionamento de volumes para NFS, iSCSI ou FC PVs neste momento. Trident suporta a expansão de volumes NFS, iSCSI e FC.

Os detalhes de conexão do PV não podem ser modificados após a criação.

Criar snapshots de volume sob demanda

Trident suporta a criação de snapshots de volumes sob demanda e a criação de PVCs a partir de snapshots usando a estrutura CSI. Os snapshots fornecem um método conveniente para manter cópias dos dados em

um ponto no tempo e têm um ciclo de vida independente do PV de origem no Kubernetes. Esses snapshots podem ser usados para clonar PVCs.

Criar volumes a partir de snapshots

Trident também suporta a criação de PersistentVolumes a partir de snapshots de volume. Para isso, basta criar um PersistentVolumeClaim e mencionar o `datasource` como o snapshot necessário a partir do qual o volume precisa ser criado. Trident gerenciará esse PVC criando um volume com os dados presentes no snapshot. Com esse recurso, é possível duplicar dados entre regiões, criar ambientes de teste, substituir um volume de produção danificado ou corrompido por completo ou recuperar arquivos e diretórios específicos e transferi-los para outro volume conectado.

Mover volumes no cluster

Os administradores de storage têm a capacidade de mover volumes entre agregados e controladores no cluster ONTAP de forma não disruptiva para o consumidor de storage. Essa operação não afeta o Trident nem o cluster Kubernetes, desde que o agregado de destino seja um ao qual a SVM que o Trident está utilizando tenha acesso. É importante ressaltar que, se o agregado foi adicionado recentemente à SVM, o backend precisará ser atualizado ao ser readicionado ao Trident. Isso fará com que o Trident faça um novo inventário da SVM para que o novo agregado seja reconhecido.

No entanto, a movimentação de volumes entre backends não é suportada automaticamente pelo Trident. Isso inclui entre SVMs no mesmo cluster, entre clusters ou para uma plataforma de storage diferente (mesmo que esse sistema de storage esteja conectado ao Trident).

Se um volume for copiado para outro local, o recurso de importação de volume pode ser usado para importar volumes atuais para Trident.

Expandir volumes

Trident suporta o redimensionamento de PVs NFS, iSCSI e FC. Isso permite que os usuários redimensionem seus volumes diretamente pela camada Kubernetes. A expansão de volumes é possível para todas as principais plataformas de storage NetApp, incluindo ONTAP, e backends SolidFire/NetApp HCI. Para permitir uma possível expansão futura, defina `allowVolumeExpansion` como `true` no seu StorageClass associado ao volume. Sempre que o Persistent Volume precisar ser redimensionado, edite a anotação `spec.resources.requests.storage` no Persistent Volume Claim para o tamanho de volume necessário. Trident cuidará automaticamente do redimensionamento do volume no cluster de storage.

Importar um volume existente ao Kubernetes

A importação de volumes permite importar um volume de armazenamento existente para um ambiente Kubernetes. Atualmente, isso é suportado pelos `ontap-nas`, `ontap-nas-flexgroup`, `solidfire-san` e `azure-netapp-files` drivers. Esse recurso é útil ao migrar um aplicativo existente para o Kubernetes ou em cenários de recuperação de desastres.

Ao usar os drivers ONTAP e `solidfire-san`, utilize o comando `tridentctl import volume <backend-name> <volume-name> -f /path/pvc.yaml` para importar um volume existente para o Kubernetes para ser gerenciado pelo Trident. O arquivo PVC YAML ou JSON usado no comando de importação de volume aponta para uma storage class que identifica o Trident como o provisionador. Ao usar um backend NetApp HCI/SolidFire, certifique-se de que os nomes dos volumes sejam únicos. Se os nomes dos volumes estiverem duplicados, clone o volume para um nome único para que o recurso de importação de volume possa distingui-los.

Se o `azure-netapp-files` driver for utilizado, use o comando `tridentctl import volume`

`<backend-name> <volume path> -f /path/pvc.yaml` para importar o volume para o Kubernetes para ser gerenciado pelo Trident. Isso garante uma referência de volume exclusiva.

Ao executar o comando acima, Trident localizará o volume no backend e lerá seu tamanho. Ele adicionará automaticamente (e sobrescreverá, se necessário) o tamanho do volume do PVC configurado. Em seguida, Trident cria o novo PV e o Kubernetes vincula o PVC ao PV.

Se um contêiner foi implantado de forma que exigisse o PVC importado específico, ele permanecerá em estado pendente até que o par PVC/PV seja vinculado por meio do processo de importação de volume. Após a vinculação do par PVC/PV, o contêiner deverá ser iniciado, desde que não haja outros problemas.

Serviço de registro

A implantação e o gerenciamento do armazenamento para o registro foram documentados em "netapp.io" no "blog".

Serviço de logging

Assim como outros OpenShift serviços, o serviço de registro de logs é implantado usando o Ansible com parâmetros de configuração fornecidos pelo arquivo de inventário, também conhecido como hosts, fornecido ao playbook. Há dois métodos de instalação que serão abordados: implantar o serviço de registro de logs durante a instalação inicial do OpenShift e implantar o serviço de registro de logs após o OpenShift ter sido instalado.

A partir da versão 3.9 do Red Hat OpenShift, a documentação oficial recomenda não utilizar NFS para o serviço de registro de logs devido a preocupações com corrupção de dados. Isso se baseia em testes realizados pela Red Hat em seus produtos. O servidor NFS do ONTAP não apresenta esses problemas e pode facilmente suportar uma implementação de registro de logs. Em última análise, a escolha do protocolo para o serviço de registro de logs é sua, apenas saiba que ambos funcionarão muito bem ao usar plataformas NetApp e não há motivo para evitar NFS se essa for sua preferência.

Se optar por usar NFS com o serviço de registro de logs, você precisará definir a variável Ansible `openshift_enable_unsupported_configurations` para `true` evitar que o instalador falhe.

Comece agora

O serviço de registro de logs pode, opcionalmente, ser implementado tanto para aplicações quanto para as operações principais do próprio cluster OpenShift. Se você optar por implementar o registro de logs de operações, especificando a variável `openshift_logging_use_ops` como `true`, duas instâncias do serviço serão criadas. As variáveis que controlam a instância de registro de logs para operações contêm "ops", enquanto a instância para aplicações não.

Configurar as variáveis do Ansible de acordo com o método de implantação é importante para garantir que o storage correto seja utilizado pelos serviços subjacentes. Vamos analisar as opções para cada um dos métodos de implantação.

As tabelas abaixo contêm apenas as variáveis relevantes para configuração de storage conforme relacionado ao serviço de registro de logs. Você pode encontrar outras opções em "Documentação de registro de logs do Red Hat OpenShift" que devem ser revisadas, configuradas e utilizadas de acordo com a sua implementação.

As variáveis na tabela abaixo farão com que o playbook do Ansible crie um PV e um PVC para o serviço de logging usando os detalhes fornecidos. Este método é significativamente menos flexível do que usar o

playbook de instalação de componentes após a instalação do OpenShift, porém, se você já tiver volumes disponíveis, é uma opção.

Variável	Detalhes
`openshift_logging_storage_kind`	Defina como `nfs` para que o instalador crie um NFS PV para o serviço de registro de logs.
`openshift_logging_storage_host`	O nome do host ou endereço IP do host NFS. Isso deve ser definido como o dataLIF da sua máquina virtual.
`openshift_logging_storage_nfs_directory`	O caminho de montagem para a exportação NFS. Por exemplo, se o volume estiver vinculado como `/openshift_logging`, você usaria esse caminho para esta variável.
`openshift_logging_storage_volume_name`	O nome, por exemplo, `pv_ose_logs`, do PV a ser criado.
`openshift_logging_storage_volume_size`	O tamanho da exportação NFS, por exemplo `100Gi`.

Se o seu OpenShift cluster já estiver em execução e, portanto, Trident já tiver sido implantado e configurado, o instalador poderá usar o provisionamento dinâmico para criar os volumes. As seguintes variáveis precisarão ser configuradas.

Variável	Detalhes
`openshift_logging_es_pvc_dynamic`	Defina como true para usar volumes provisionados dinamicamente.
`openshift_logging_es_pvc_storage_class_name`	O nome da storage class que será utilizada no PVC.
`openshift_logging_es_pvc_size`	O tamanho do volume solicitado no PVC.
`openshift_logging_es_pvc_prefix`	Um prefixo para os PVCs usados pelo serviço de logging.
`openshift_logging_es_ops_pvc_dynamic`	Defina como `true` para usar volumes provisionados dinamicamente para a instância de logging de operações.
`openshift_logging_es_ops_pvc_storage_class_name`	O nome da storage class para a instância de ops logging.
`openshift_logging_es_ops_pvc_size`	O tamanho da solicitação de volume para a instância ops.
`openshift_logging_es_ops_pvc_prefix`	Um prefixo para os PVCs da instância ops.

Implante a pilha de registro de logs

Se você estiver implementando o registro de logs como parte do processo de instalação inicial do OpenShift, basta seguir o processo de implantação padrão. O Ansible configurará e implantará os serviços e objetos do OpenShift necessários para que o serviço esteja disponível assim que o Ansible for concluído.

No entanto, se você estiver realizando a implantação após a instalação inicial, o playbook do componente precisará ser usado pelo Ansible. Esse processo pode variar um pouco com diferentes versões do OpenShift,

portanto, certifique-se de ler e seguir "Documentação do Red Hat OpenShift Container Platform 3.11" para a sua versão.

Serviço de métricas

O serviço de métricas fornece informações valiosas ao administrador sobre o status, a utilização de recursos e a disponibilidade do OpenShift cluster. Ele também é necessário para a funcionalidade de escalonamento automático de pods e muitas organizações utilizam dados do serviço de métricas para seus aplicativos de cobrança e/ou demonstração de custos.

Assim como no serviço de logging, e OpenShift como um todo, Ansible é usado para implantar o serviço de métricas. Também, como o serviço de logging, o serviço de métricas pode ser implantado durante a configuração inicial do cluster ou após sua operação, utilizando o método de instalação de componentes. As tabelas a seguir contêm as variáveis que são importantes ao configurar storage persistente para o serviço de métricas.

As tabelas abaixo contêm apenas as variáveis relevantes para configuração de storage conforme relacionado ao serviço de métricas. Existem muitas outras opções encontradas na documentação que devem ser revisadas, configuradas e utilizadas de acordo com a sua implementação.

Variável	Detalhes
`openshift_metrics_storage_kind`	Defina como `nfs` para que o instalador crie um NFS PV para o serviço de registro de logs.
`openshift_metrics_storage_host`	O nome do host ou endereço IP do host NFS. Este valor deve ser definido como o dataLIF da sua SVM.
`openshift_metrics_storage_nfs_directory`	O caminho de montagem para a exportação NFS. Por exemplo, se o volume estiver vinculado como `/openshift_metrics`, você usaria esse caminho para esta variável.
`openshift_metrics_storage_volume_name`	O nome, por exemplo, `pv_ose_metrics`, do PV a ser criado.
`openshift_metrics_storage_volume_size`	O tamanho da exportação NFS, por exemplo `100Gi`.

Se o seu OpenShift cluster já estiver em execução e, portanto, Trident já tiver sido implantado e configurado, o instalador poderá usar o provisionamento dinâmico para criar os volumes. As seguintes variáveis precisarão ser configuradas.

Variável	Detalhes
`openshift_metrics_cassandra_pvc_prefix`	Um prefixo a ser usado para os PVCs de métricas.
`openshift_metrics_cassandra_pvc_size`	O tamanho dos volumes a serem solicitados.
`openshift_metrics_cassandra_storage_type`	O tipo de storage a ser usado para métricas, isso deve ser definido como dinâmico para que o Ansible crie PVCs com a classe de storage apropriada.
`openshift_metrics_cassanda_pvc_storage_class_name`	O nome da storage class a ser usada.

Implante o serviço de métricas

Com as variáveis Ansible apropriadas definidas no seu arquivo hosts/inventory, implante o serviço usando Ansible. Se você estiver implantando no momento da instalação do OpenShift, o PV será criado e usado automaticamente. Se estiver implantando usando os playbooks de componentes, após a instalação do OpenShift, o Ansible criará todos os PVCs necessários e, depois que Trident provisionar storage para eles, implantará o serviço.

As variáveis acima e o processo de implantação podem mudar a cada versão do OpenShift. Certifique-se de revisar e seguir "Guia de implantação da Red Hat's OpenShift" para a sua versão para que esteja configurado para o seu ambiente.

Proteção de dados e recuperação de desastres

Saiba mais sobre as opções de proteção e recuperação para Trident e volumes criados usando Trident. Você deve ter uma estratégia de proteção e recuperação de dados para cada aplicação com requisito de persistência.

Replicação e recuperação do Trident

Você pode criar um backup para restaurar Trident em caso de desastre.

Trident replicação

Trident usa CRDs do Kubernetes para armazenar e gerenciar seu próprio estado e o etcd do cluster Kubernetes para armazenar seus metadados.

Passos

1. Faça backup do cluster Kubernetes etcd usando "Kubernetes: fazendo backup de um cluster etcd".
2. Coloque os artefatos de backup em um FlexVol volume

> NetApp recomenda que você proteja a SVM onde o FlexVol reside com uma relação de SnapMirror para outra SVM.

Recuperação do Trident

Utilizando CRDs do Kubernetes e o snapshot do etcd do cluster Kubernetes, você pode recuperar Trident.

Passos

1. A partir da SVM de destino, monte o volume que contém os arquivos de dados e certificados do Kubernetes etcd no host que será configurado como nó mestre.
2. Copie todos os certificados necessários referentes ao cluster Kubernetes em `/etc/kubernetes/pki` e os arquivos de membros do etcd em `/var/lib/etcd`.
3. Restaure o cluster Kubernetes a partir do backup etcd usando "Kubernetes: restaurando um cluster etcd".
4. Execute `kubectl get crd` para verificar se todos os recursos personalizados do Trident foram iniciados e recupere os objetos do Trident para verificar se todos os dados estão disponíveis.

Replicação e recuperação de SVM

Trident não consegue configurar relações de replicação, no entanto, o administrador de storage pode usar "ONTAP SnapMirror" para replicar uma SVM.

Em caso de desastre, você pode ativar a SVM de destino do SnapMirror para começar a fornecer dados. Você pode retornar à primária quando os sistemas forem restaurados.

Sobre esta tarefa

Considere o seguinte ao usar o recurso de Replicação SVM do SnapMirror:

- Você deve criar um backend distinto para cada SVM com SVM-DR habilitado.
- Configure as classes de armazenamento para selecionar apenas os backends replicados quando necessário, para evitar que volumes que não precisam de replicação sejam provisionados nos backends que suportam SVM-DR.
- Os administradores de aplicativos devem compreender os custos e a complexidade adicionais associados à replicação e considerar cuidadosamente seu plano de recuperação antes de iniciar este processo.

Replicação SVM

Você pode usar "ONTAP: SnapMirror SVM replicação" para criar a relação de replicação SVM.

SnapMirror permite definir opções para controlar o que replicar. Você precisará saber quais opções selecionou ao executar Recuperação de SVM usando Trident.

- "-identity-preserve true" replica toda a configuração da SVM.
- "-descartar-configs network" Exclui LIFs e configurações de rede relacionadas.
- "-identity-preserve false" replica apenas os volumes e a configuração de segurança.

Recuperação de SVM usando Trident

Trident não detecta automaticamente falhas de SVM. Em caso de desastre, o administrador pode iniciar manualmente o failover do Trident para a nova SVM.

Passos

1. Cancele as transferências SnapMirror agendadas e em andamento, interrompa a relação de replicação, pare o SVM de origem e, em seguida, ative o SVM de destino SnapMirror.
2. Se você especificou `-identity-preserve false` ou `-discard-config network` ao configurar a replicação do SVM, atualize o `managementLIF` e o `dataLIF` no arquivo de definição do backend do Trident.
3. Confirme `storagePrefix` está presente no arquivo de definição do backend Trident. Este parâmetro não pode ser alterado. Omitir `storagePrefix` fará com que a atualização do backend falhe.
4. Atualize todos os backends necessários para refletir o novo nome da SVM de destino usando:

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

5. Se você especificou `-identity-preserve false` ou `discard-config network`, você deve reiniciar todos os pods do aplicativo.

Se você especificou `-identity-preserve true`, todos os volumes provisionados pelo Trident começam a fornecer dados quando o SVM de destino é ativado.

Replicação e recuperação de volume

Trident não consegue configurar relações de replicação do SnapMirror, no entanto, o administrador de storage pode usar "ONTAP SnapMirror replicação e recuperação" para replicar volumes criados pelo Trident.

Em seguida, você pode importar os volumes recuperados para Trident usando "tridentctl volume import".

A importação não é suportada em `ontap-nas-economy`, `ontap-san-economy` ou `ontap-flexgroup-economy` drivers.

Proteção de dados Snapshot

Você pode proteger e restaurar dados usando:

- Um controlador de snapshot externo e CRDs para criar snapshots de Persistent Volumes (PVs) do Kubernetes.

  "Instantâneos de volume"

- ONTAP Snapshots para restaurar todo o conteúdo de um volume ou recuperar arquivos ou LUNs individuais.

  "Snapshots do ONTAP"

Automatizando o failover de aplicações stateful com Trident

O recurso de desanexação forçada do Trident permite desanexar automaticamente volumes de nós com problemas em um cluster Kubernetes, evitando corrupção de dados e garantindo a disponibilidade do aplicativo. Esse recurso é particularmente útil em cenários onde os nós ficam inativos ou são colocados offline para manutenção.

Detalhes sobre force detach

A desanexação forçada está disponível para `ontap-san`, `ontap-san-economy`, `ontap-nas` e `ontap-nas-economy` apenas. Antes de habilitar a desanexação forçada, o desligamento não gradual do nó (NGNS) deve estar habilitado no cluster Kubernetes. O NGNS está habilitado por padrão para Kubernetes 1.28 e superiores. Para obter mais informações, consulte "Kubernetes: encerramento abrupto de nós".

Ao usar o `ontap-nas` ou `ontap-nas-economy` driver, você precisa definir o parâmetro `autoExportPolicy` na configuração do backend para `true` que o Trident possa restringir o acesso do nó Kubernetes com o taint aplicado usando políticas de exportação gerenciadas.

Como Trident depende do NGNS do Kubernetes, não remova `out-of-service` taints de um nó não íntegro até que todas as cargas de trabalho não toleráveis sejam reagendadas. Aplicar ou remover a taint de forma imprudente pode comprometer a proteção de dados do backend.

Quando o administrador do cluster Kubernetes aplicou a `node.kubernetes.io/out-of-service=nodeshutdown:NoExecute` taint ao nó e `enableForceDetach` está definida como `true`, Trident determinará o status do nó e:

1. Interrompa o acesso de E/S de backend para volumes montados nesse nó.
2. Marque o objeto do nó Trident como `dirty` (não seguro para novas publicações).

> ⓘ O controlador Trident rejeitará novas solicitações de publicação de volume até que o nó seja requalificado (após ter sido marcado como `dirty`) pelo pod do nó Trident. Quaisquer cargas de trabalho agendadas com um PVC montado (mesmo depois que o nó de cluster estiver íntegro e pronto) não serão aceitas até que Trident possa verificar o nó `clean` (seguro para novas publicações).

Quando a integridade do nó for restaurada e a taint for removida, Trident irá:

1. Identifique e limpe caminhos publicados obsoletos no nó.
2. Se o nó estiver em um `cleanable` estado (a marcação de fora de serviço foi removida e o nó está em `Ready` estado) e todos os caminhos publicados obsoletos estiverem limpos, Trident readmitirá o nó como `clean` e permitirá novos volumes publicados no nó.

Detalhes sobre failover automatizado

Você pode automatizar o processo de desanexação forçada por meio da integração com "operador de verificação de integridade do nó (NHC)". Quando ocorre uma falha em um nó, o NHC aciona a remediação de nó do Trident (TNR) e a desanexação forçada automaticamente, criando um CR TridentNodeRemediation no namespace do Trident definindo o nó com falha. O TNR é criado somente após a falha do nó e removido pelo NHC assim que o nó volta a ficar online ou é excluído.

Falha no processo de remoção do pod do nó

O failover automatizado seleciona as cargas de trabalho a serem removidas do nó com falha. Quando um TNR é criado, o controlador TNR marca o nó como sujo, impedindo novas publicações de volume e começa a remover os pods compatíveis com desanexação forçada e seus anexos de volume.

Todos os volumes/PVCs suportados por force-detach são suportados por failover automatizado:

- NAS, e volumes NAS-economy usando políticas de exportação automática (SMB ainda não é suportado).
- SAN e volumes SAN-economy.

Consulte Detalhes sobre force detach.

Comportamento padrão:

- Os pods que utilizam volumes compatíveis com o recurso de desanexação forçada são removidos do nó com falha. O Kubernetes irá reagendá-los para um nó íntegro.
- Os pods que utilizam um volume não suportado pelo force-detach, incluindo volumes não-Trident, não são removidos do nó com falha.
- Os pods sem estado (não PVCs) não são removidos do nó com falha, a menos que a anotação do pod `trident.netapp.io/podRemediationPolicy: delete` esteja definida.

Substituindo o comportamento de remoção do pod:

O comportamento de remoção de pods pode ser personalizado usando uma anotação de pod: `trident.netapp.io/podRemediationPolicy[retain, delete]`. Essas anotações são examinadas e usadas quando ocorre um failover. Aplique anotações à especificação do pod do deployment/replicaset do Kubernetes para evitar que a anotação desapareça após um failover:

- `retain` - O pod NÃO será removido do nó com falha durante um failover automatizado.
- `delete` - O Pod será removido do nó com falha durante um failover automatizado.

Essas anotações podem ser aplicadas a qualquer pod.

- As operações de I/O serão bloqueadas apenas em nós com falha para volumes que suportam force-detach.
- Para volumes que não suportam force-detach, existe o risco de corrupção de dados e problemas de multi-attach.

CR TridentNodeRemediation

O TridentNodeRemediation (TNR) CR define um nó com falha. O nome do TNR é o nome do nó com falha.

Exemplo de TNR:

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

TNR states: Use os seguintes comandos para visualizar o status dos TNRs:
`kubectl get tnr <name> -n <trident-namespace>`

Os TNRs podem estar em um dos seguintes estados:

- *Remediando*:
  - Cessar o acesso de E/S de backend para volumes suportados por force-detach montados nesse nó.
  - O objeto de nó Trident está marcado como sujo (não é seguro para novas publicações).
  - Remova os pods e os anexos de volume do nó
- *Recuperação de nó pendente*:
  - O controlador está aguardando que o nó volte a ficar online.
  - Assim que o nó estiver online, publish-enforcement garantirá que o nó esteja limpo e pronto para novas publicações de volume.
- Se o nó for excluído do K8s, o controlador TNR removerá o TNR e interromperá a reconciliação.
- *Concluído*:
  - Todas as etapas de remediação e recuperação do nó foram concluídas com sucesso. O nó está limpo e pronto para novas publicações de volume.
- *Fracassado*:

- Erro irrecuperável. Os motivos do erro são definidos no campo status.message do CR.

Habilitando o failover automatizado

Pré-requisitos:

- Certifique-se de que o recurso de desanexação forçada esteja ativado antes de ativar o failover automatizado. Para obter mais informações, consulte Detalhes sobre force detach.
- Instale a verificação de integridade (NHC) no cluster Kubernetes.
  - "Instalar o operator-sdk".
  - Instale o Operator Lifecycle Manager (OLM) no cluster se ainda não estiver instalado: `operator-sdk olm install`.
  - Instale o operador de verificação de integridade do nó: `kubectl create -f https://operatorhub.io/install/node-healthcheck-operator.yaml`.

> Você também pode usar métodos alternativos para detectar falha de nó, conforme especificado na [Integrating Custom Node Health Check Solutions] seção abaixo.

Consulte "Operador de verificação de integridade do nó" para mais informações.

Passos

1. Crie um NodeHealthCheck (NHC) no namespace Trident para monitorar os nós de trabalho no cluster. Exemplo:

```
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. Aplique o CR de verificação de integridade do nó no `trident` namespace.

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

A configuração CR acima monitora os nós de trabalho do K8s em busca das condições de nó Ready: false e Unknown. O failover automatizado será acionado quando um nó entrar no estado Ready: false ou Ready: Unknown.

O `unhealthyConditions` no CR usa um período de carência de 0 segundos. Isso faz com que o failover automatizado seja acionado imediatamente após o K8s definir a condição do nó como Ready: false, que é definida depois que o K8s perde o heartbeat de um nó. O K8s tem um padrão de espera de 40 segundos após o último heartbeat antes de definir Ready: false. Esse período de carência pode ser personalizado nas opções de implantação do K8s.

Para opções de configuração adicionais, consulte "Documentação do Node-Healthcheck-Operator".

Informações adicionais de configuração

Quando Trident é instalado com o recurso de desanexação forçada ativado, dois recursos adicionais são criados automaticamente no namespace do Trident para facilitar a integração com NHC: TridentNodeRemediationTemplate (TNRT) e ClusterRole.

TridentNodeRemediationTemplate (TNRT):

O TNRT serve como modelo para o controlador NHC, que usa TNRT para gerar recursos TNR conforme necessário.

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

ClusterRole:

Uma função de cluster também é adicionada durante a instalação quando force-detach está habilitado. Isso concede permissões ao NHC para os TNRs no 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
```

Atualizações e manutenção de cluster K8s

Para evitar qualquer failover, pause o failover automatizado durante a manutenção ou atualizações do K8s, quando se espera que os nós fiquem indisponíveis ou reinicializem. Você pode pausar o CR do NHC (descrito acima) aplicando um patch no seu CR:

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

Isso pausa o failover automatizado. Para reativar o failover automatizado, remova o pauseRequests da especificação após a conclusão da manutenção.

Limitações

- As operações de E/S são bloqueadas apenas nos nós com falha para volumes compatíveis com force-detach. Somente os pods que utilizam volumes/PVCs compatíveis com force-detach são removidos automaticamente.
- O failover automatizado e o force-detach são executados dentro do pod trident-controller. Se o nó que hospeda o trident-controller falhar, o failover automatizado será atrasado até que o K8s mova o pod para um nó saudável.

Integrando soluções personalizadas de verificação de integridade de nós

Você pode substituir o Node Healthcheck Operator por ferramentas alternativas de detecção de falhas de nó para acionar o failover automatizado. Para garantir a compatibilidade com o mecanismo de failover automatizado, sua solução personalizada deve:

- Crie um TNR quando uma falha de nó for detectada, usando o nome do nó com falha como o nome do CR do TNR.
- Exclua o TNR quando o nó se recuperar e o TNR estiver no estado Succeeded.

Segurança

Segurança

Utilize as recomendações listadas aqui para garantir que sua instalação do Trident seja segura.

Execute Trident em seu próprio namespace

É importante impedir que aplicativos, administradores de aplicativos, usuários e aplicativos de gerenciamento acessem as definições de objetos do Trident ou os pods para garantir um armazenamento confiável e bloquear possíveis atividades maliciosas.

Para separar os outros aplicativos e usuários do Trident, sempre instale Trident em seu próprio namespace do Kubernetes (`trident`). Colocar Trident em seu próprio namespace garante que apenas a equipe administrativa do Kubernetes tenha acesso ao pod do Trident e aos artefatos (como segredos de backend e CHAP, se aplicável) armazenados nos objetos CRD do namespace. Você deve garantir que somente os administradores tenham acesso ao namespace do Trident e, assim, acesso ao `tridentctl` aplicativo.

Use autenticação CHAP com backends SAN do ONTAP

Trident oferece suporte à autenticação baseada em CHAP para cargas de trabalho ONTAP SAN (usando os drivers `ontap-san` e `ontap-san-economy`). NetApp recomenda o uso de CHAP bidirecional com Trident para autenticação entre um host e o backend de storage.

Para backends ONTAP que utilizam os drivers de armazenamento SAN, Trident pode configurar o CHAP bidirecional e gerenciar nomes de usuário e segredos CHAP por meio de `tridentctl`. Consulte "Prepare-se para configurar o backend com os drivers ONTAP SAN" para entender como o Trident configura o CHAP em backends ONTAP.

Use autenticação CHAP com NetApp HCI e SolidFire backends

NetApp recomenda implantar CHAP bidirecional para garantir autenticação entre um host e os backends NetApp HCI e SolidFire. Trident usa um objeto secreto que inclui duas senhas CHAP por tenant. Quando Trident é instalado, ele gerencia os segredos CHAP e os armazena em um `tridentvolume` objeto CR para o respectivo PV. Quando você cria um PV, Trident usa os segredos CHAP para iniciar uma sessão iSCSI e se comunicar com o sistema NetApp HCI e SolidFire via CHAP.

Os volumes criados pelo Trident não estão associados a nenhum Volume Access Group.

Use Trident com NVE e NAE

NetApp ONTAP fornece criptografia de dados em repouso para proteger dados confidenciais caso um disco seja roubado, devolvido ou reutilizado. Para detalhes, consulte "Configurar visão geral da criptografia de volume NetApp".

- Se o NAE estiver habilitado no backend, qualquer volume provisionado no Trident será habilitado para NAE.
  - Você pode definir o sinalizador de criptografia NVE para `""` criar volumes com NAE habilitado.
- Se o NAE não estiver habilitado no backend, qualquer volume provisionado no Trident terá o NVE habilitado, a menos que o sinalizador de criptografia NVE esteja definido como `false` (o valor padrão) na configuração do backend.

Volumes criados no Trident em um backend habilitado para NAE devem ser criptografados com NVE ou NAE.

- Você pode definir o sinalizador de criptografia NVE `true` na configuração do backend Trident para substituir a criptografia NAE e usar uma chave de criptografia específica para cada volume.
- Definir o sinalizador de criptografia NVE para `false` em um backend com NAE habilitado cria um volume com NAE habilitado. Você não pode desabilitar a criptografia NAE definindo o sinalizador de criptografia NVE para `false`.

- Você pode criar manualmente um volume NVE no Trident definindo explicitamente o sinalizador de criptografia NVE como `true`.

Para obter mais informações sobre as opções de configuração do backend, consulte:

- "Opções de configuração do ONTAP SAN"
- "Opções de configuração do ONTAP NAS"

Linux Unified Key Setup (LUKS)

Você pode habilitar o Linux Unified Key Setup (LUKS) para criptografar volumes ONTAP SAN e ONTAP SAN ECONOMY no Trident. Trident oferece suporte à rotação de senha e expansão de volume para volumes criptografados com LUKS.

No Trident, os volumes criptografados com LUKS usam a cifra e o modo aes-xts-plain64, conforme recomendado por "NIST".

A criptografia LUKS não é compatível com sistemas ASA r2. Para obter informações sobre sistemas ASA r2, consulte "Saiba mais sobre os sistemas de armazenamento ASA r2".

Antes de começar

- Os nós de trabalho devem ter cryptsetup 2.1 ou superior (mas inferior a 3.0) instalado. Para mais informações, visite "Gitlab: cryptsetup".
- Por motivos de desempenho, NetApp recomenda que os nós de trabalho suportem Advanced Encryption Standard New Instructions (AES-NI). Para verificar o suporte a AES-NI, execute o seguinte comando:

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

Se nada for retornado, seu processador não suporta AES-NI. Para mais informações sobre AES-NI, visite: "Intel: instruções do Padrão de Criptografia Avançada (AES-NI)".

Ativar criptografia LUKS

Você pode habilitar a criptografia por volume, do lado do host, usando o Linux Unified Key Setup (LUKS) para volumes ONTAP SAN e ONTAP SAN ECONOMY.

Passos

1. Defina os atributos de criptografia LUKS na configuração do backend. Para obter mais informações sobre as opções de configuração do backend para ONTAP SAN, consulte "Opções de configuração do ONTAP SAN".

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

2. Use `parameters.selector` para definir os pools de armazenamento usando criptografia LUKS. Por exemplo:

```
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. Crie um segredo que contenha a senha LUKS. Por exemplo:

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

Limitações

Volumes criptografados com LUKS não podem aproveitar a deduplicação e a compressão do ONTAP.

Configuração de backend para importação de volumes LUKS

Para importar um volume LUKS, você deve definir `luksEncryption` para (`true` no backend. A opção `luksEncryption` informa ao Trident se o volume é compatível com LUKS (`true` ou não compatível com LUKS (`false`, conforme mostrado no exemplo a seguir.

```
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'
```

Configuração de PVC para importação de volumes LUKS

Para importar volumes LUKS dinamicamente, defina a anotação `trident.netapp.io/luksEncryption` para `true` e inclua uma classe de armazenamento habilitada para LUKS no PVC, conforme mostrado neste exemplo.

```
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
```

Rotacionar uma frase secreta LUKS

Você pode alternar a senha LUKS e confirmar a rotação.

> Não se esqueça de uma senha até verificar se ela não está mais sendo referenciada por nenhum volume, snapshot ou segredo. Se uma senha referenciada for perdida, você poderá não conseguir montar o volume e os dados permanecerão criptografados e inacessíveis.

Sobre esta tarefa

A rotação da senha LUKS ocorre quando um pod que monta o volume é criado após a especificação de uma nova senha LUKS. Quando um novo pod é criado, Trident compara a senha LUKS no volume com a senha ativa no segredo.

- Se a senha no volume não corresponder à senha ativa no segredo, a rotação ocorrerá.
- Se a senha no volume corresponder à senha ativa no segredo, o `previous-luks-passphrase` parâmetro será ignorado.

Passos

1. Adicione os `node-publish-secret-name` e `node-publish-secret-namespace` parâmetros StorageClass. Por exemplo:

```
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. Identifique as frases secretas existentes no volume ou no 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. Atualize o segredo LUKS do volume para especificar as senhas nova e anterior. Certifique-se de que `previous-luke-passphrase-name` e `previous-luks-passphrase` correspondam à senha anterior.

```
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. Crie um novo pod montando o volume. Isso é necessário para iniciar a rotação.
5. Verifique se a senha foi rotacionada.

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"]
```

Resultados

A senha foi rotacionada quando apenas a nova senha foi retornada no volume e no snapshot.

> Se duas senhas forem retornadas, por exemplo `luksPassphraseNames: ["B", "A"]`, a rotação estará incompleta. Você pode acionar um novo pod para tentar concluir a rotação.

Habilitar expansão de volume

Você pode habilitar a expansão de volume em um volume criptografado com LUKS.

Passos

1. Ative o `CSINodeExpandSecret`feature gate (beta 1.25+). Consulte "Kubernetes 1.25: use segredos para expansão de volumes CSI orientada a nós" para obter detalhes.
2. Adicione os `node-expand-secret-name` e `node-expand-secret-namespace` parâmetros StorageClass. Por exemplo:

```
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
```

Resultados

Ao iniciar a expansão de storage online, o kubelet passa as credenciais apropriadas para o driver.

Criptografia em trânsito Kerberos

Usando criptografia em trânsito Kerberos, você pode melhorar a segurança do acesso aos dados habilitando a criptografia para o tráfego entre seu cluster gerenciado e o storage backend.

Trident oferece suporte à criptografia Kerberos para ONTAP como backend de storage:

- **On-premise ONTAP** - Trident oferece suporte à criptografia Kerberos em conexões NFSv3 e NFSv4 de clusters Red Hat OpenShift e Kubernetes upstream para volumes ONTAP locais.

Você pode criar, excluir, redimensionar, criar snapshots, clonar, clonar em modo somente leitura e importar volumes que utilizam criptografia NFS.

Configurar criptografia Kerberos em trânsito com volumes ONTAP locais

Você pode habilitar a criptografia Kerberos no tráfego de storage entre seu cluster gerenciado e um backend de storage ONTAP local.

> ⓘ A criptografia Kerberos para o tráfego NFS com backends de storage ONTAP locais só é compatível usando o driver de storage `ontap-nas`.

Antes de começar

- Certifique-se de ter acesso ao utilitário `tridentctl`.
- Certifique-se de ter acesso de administrador ao backend de storage ONTAP.
- Certifique-se de saber o nome do(s) volume(s) que você compartilhará do backend de storage ONTAP.
- Certifique-se de ter preparado a máquina virtual de storage ONTAP para suportar a criptografia Kerberos para volumes NFS. Consulte "Habilite o Kerberos em um dataLIF" para obter instruções.
- Certifique-se de que todos os volumes NFSv4 que você usa com criptografia Kerberos estejam configurados corretamente. Consulte a seção Configuração de Domínio NFSv4 da NetApp (página 13) do "NetApp NFSv4 Melhorias e Guia de Práticas Recomendadas".

Adicionar ou modificar políticas de exportação do ONTAP

Você precisa adicionar regras às políticas de exportação ONTAP existentes ou criar novas políticas de exportação que ofereçam suporte à criptografia Kerberos para o volume raiz da máquina virtual de storage ONTAP, bem como para quaisquer volumes ONTAP compartilhados com o cluster Kubernetes upstream. As regras de política de exportação que você adicionar, ou as novas políticas de exportação que você criar, precisam oferecer suporte aos seguintes protocolos de acesso e permissões de acesso:

Protocolos de acesso

Configure a política de exportação com os protocolos de acesso NFS, NFSv3 e NFSv4.

Detalhes de acesso

Você pode configurar uma das três versões diferentes de criptografia Kerberos, dependendo das suas necessidades para o volume:

- **Kerberos 5** - (autenticação e criptografia)
- **Kerberos 5i** - (autenticação e criptografia com proteção de identidade)
- **Kerberos 5p** - (autenticação e criptografia com proteção de identidade e privacidade)

Configure a regra de política de exportação do ONTAP com as permissões de acesso apropriadas. Por exemplo, se os clusters forem montar os volumes NFS com uma combinação de criptografia Kerberos 5i e Kerberos 5p, use as seguintes configurações de acesso:

Tipo	Acesso somente leitura	Acesso de leitura/gravação	Acesso de superusuário
UNIX	Habilitado	Habilitado	Habilitado
Kerberos 5i	Habilitado	Habilitado	Habilitado
Kerberos 5p	Habilitado	Habilitado	Habilitado

Consulte a seguinte documentação para saber como criar políticas de exportação do ONTAP e regras de políticas de exportação:

- "Criar uma política de exportação"
- "Adicionar uma regra a uma política de exportação"

Crie um backend de storage

Você pode criar uma configuração de backend de storage Trident que inclua a capacidade de criptografia Kerberos.

Sobre esta tarefa

Ao criar um arquivo de configuração de storage backend que configura a criptografia Kerberos, você pode especificar uma das três versões diferentes de criptografia Kerberos usando o `spec.nfsMountOptions` parâmetro:

- `spec.nfsMountOptions: sec=krb5` (autenticação e criptografia)
- `spec.nfsMountOptions: sec=krb5i` (autenticação e criptografia com proteção de identidade)
- `spec.nfsMountOptions: sec=krb5p` (autenticação e criptografia com proteção de identidade e privacidade)

Especifique apenas um nível Kerberos. Se você especificar mais de um nível de criptografia Kerberos na lista de parâmetros, somente a primeira opção será usada.

Passos

1. No cluster gerenciado, crie um arquivo de configuração de backend de storage usando o exemplo a seguir. Substitua os valores entre colchetes <> com as informações do seu 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. Use o arquivo de configuração que você criou na etapa anterior para criar o backend:

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

Se a criação do backend falhar, há algo errado com a configuração do backend. Você pode visualizar os logs para determinar a causa executando o seguinte comando:

```
tridentctl logs
```

Após identificar e corrigir o problema com o arquivo de configuração, você pode executar o comando create novamente.

Crie uma storage class

Você pode criar uma classe de armazenamento para provisionar volumes com criptografia Kerberos.

Sobre esta tarefa

Ao criar um objeto de classe de armazenamento, você pode especificar uma das três versões diferentes de criptografia Kerberos usando o `mountOptions`parâmetro:

- `mountOptions: sec=krb5` (autenticação e criptografia)
- `mountOptions: sec=krb5i` (autenticação e criptografia com proteção de identidade)
- `mountOptions: sec=krb5p` (autenticação e criptografia com proteção de identidade e privacidade)

Especifique apenas um nível Kerberos. Se você especificar mais de um nível de criptografia Kerberos na lista de parâmetros, somente a primeira opção será usada. Se o nível de criptografia que você especificou na configuração do backend de storage for diferente do nível que você especificar no objeto da classe de storage, o objeto da classe de storage terá precedência.

Passos

1. Crie um objeto Kubernetes StorageClass, usando o seguinte exemplo:

```
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. Crie a classe de armazenamento:

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

3. Certifique-se de que a classe de armazenamento foi criada:

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

   Você deverá ver uma saída semelhante à seguinte:

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

Provisionar volumes

Após criar um backend de armazenamento e uma classe de armazenamento, você pode provisionar um volume. Para obter instruções, consulte "Provisionar um volume".

Configurar criptografia Kerberos em trânsito com volumes do Azure NetApp Files

Você pode habilitar a criptografia Kerberos no tráfego de armazenamento entre seu cluster gerenciado e um único back-end de armazenamento Azure NetApp Files ou um pool virtual de back-ends de armazenamento Azure NetApp Files.

Antes de começar

- Certifique-se de ter habilitado Trident no cluster Red Hat OpenShift gerenciado.
- Certifique-se de ter acesso ao utilitário `tridentctl`.
- Certifique-se de ter preparado o backend de armazenamento Azure NetApp Files para criptografia Kerberos, observando os requisitos e seguindo as instruções em "Documentação do Azure NetApp Files".
- Certifique-se de que todos os volumes NFSv4 que você usa com criptografia Kerberos estejam configurados corretamente. Consulte a seção Configuração de Domínio NFSv4 da NetApp (página 13) do "NetApp NFSv4 Melhorias e Guia de Práticas Recomendadas".

Crie um backend de storage

Você pode criar uma configuração de back-end de armazenamento do Azure NetApp Files que inclua a capacidade de criptografia Kerberos.

Sobre esta tarefa

Ao criar um arquivo de configuração de backend de storage que configura a criptografia Kerberos, você pode defini-lo para que seja aplicado em um dos dois níveis possíveis:

- O **nível de backend de armazenamento** usando o `spec.kerberos` campo
- O **nível do pool virtual** usando o `spec.storage.kerberos` campo

Ao definir a configuração no nível do pool virtual, o pool é selecionado usando o rótulo na storage class.

Em qualquer um dos níveis, você pode especificar uma das três versões diferentes de criptografia Kerberos:

- `kerberos: sec=krb5` (autenticação e criptografia)
- `kerberos: sec=krb5i` (autenticação e criptografia com proteção de identidade)
- `kerberos: sec=krb5p` (autenticação e criptografia com proteção de identidade e privacidade)

Passos

1. No cluster gerenciado, crie um arquivo de configuração de backend de storage usando um dos exemplos a seguir, dependendo de onde você precisa definir o backend de storage (nível do backend de storage ou nível do pool virtual). Substitua os valores entre colchetes <> com as informações do seu ambiente:

Exemplo de nível de storage backend

```
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
```

Exemplo de nível de pool virtual

```
---
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. Use o arquivo de configuração que você criou na etapa anterior para criar o backend:

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

Se a criação do backend falhar, há algo errado com a configuração do backend. Você pode visualizar os logs para determinar a causa executando o seguinte comando:

```
tridentctl logs
```

Após identificar e corrigir o problema com o arquivo de configuração, você pode executar o comando create novamente.

Crie uma storage class

Você pode criar uma classe de armazenamento para provisionar volumes com criptografia Kerberos.

Passos

1. Crie um objeto Kubernetes StorageClass, usando o seguinte exemplo:

```
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. Crie a classe de armazenamento:

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

3. Certifique-se de que a classe de armazenamento foi criada:

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

Você deverá ver uma saída semelhante à seguinte:

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

Provisionar volumes

Após criar um backend de armazenamento e uma classe de armazenamento, você pode provisionar um volume. Para obter instruções, consulte "Provisionar um volume".

Informações sobre direitos autorais

Informações sobre marcas comerciais