모범 사례 및 권장 사항

Trident

NetApp
January 15, 2026

목차

모범 사례 및 권장 사항

전개

Trident 배포할 때 여기에 나열된 권장 사항을 사용하세요.

전용 네임스페이스에 배포

"네임스페이스"서로 다른 애플리케이션 간에 관리적 분리를 제공하고 리소스 공유에 대한 장벽이 됩니다. 예를 들어, 한 네임스페이스의 PVC는 다른 네임스페이스에서 사용될 수 없습니다. Trident Kubernetes 클러스터의 모든 네임스페이스에 PV 리소스를 제공하고 결과적으로 권한이 상승된 서비스 계정을 활용합니다.

또한, Trident 포드에 접근하면 사용자가 스토리지 시스템 자격 증명 및 기타 민감한 정보에 접근할 수 있습니다. 애플리케이션 사용자와 관리 애플리케이션이 Trident 객체 정의나 포드 자체에 액세스할 수 없도록 하는 것이 중요합니다.

할당량과 범위 제한을 사용하여 저장소 소비를 제어합니다.

쿠버네티스에는 두 가지 기능이 있는데, 이 두 가지를 결합하면 애플리케이션의 리소스 소비를 제한하는 강력한 메커니즘을 제공합니다. 그만큼 "저장 할당량 메커니즘" 관리자가 네임스페이스별로 글로벌 및 스토리지 클래스별 용량 및 개체 수 소비 제한을 구현할 수 있도록 합니다. 또한, 다음을 사용하여 "범위 제한" 요청이 프로비저너로 전달되기 전에 PVC 요청이 최소값과 최대값 내에 있는지 확인합니다.

이러한 값은 네임스페이스별로 정의됩니다. 즉, 각 네임스페이스에는 리소스 요구 사항에 맞는 값이 정의되어 있어야 합니다. 여기에서 정보를 확인하세요 "할당량을 활용하는 방법" .

스토리지 구성

NetApp 포트폴리오의 각 스토리지 플랫폼은 컨테이너화 여부와 관계없이 애플리케이션에 도움이 되는 고유한 기능을 갖추고 있습니다.

플랫폼 개요

Trident ONTAP 및 Element와 함께 작동합니다. 모든 애플리케이션과 시나리오에 더 적합한 플랫폼은 없습니다. 그러나 플랫폼을 선택할 때는 애플리케이션의 요구 사항과 장치를 관리하는 팀을 고려해야 합니다.

사용하는 프로토콜에 맞는 호스트 운영 체제의 기본 모범 사례를 따라야 합니다. 선택적으로, 특정 애플리케이션에 맞게 스토리지를 최적화하기 위해 백엔드, 스토리지 클래스 및 PVC 설정과 함께 애플리케이션 모범 사례를 통합하는 것을 고려할 수도 있습니다.

ONTAP 및 Cloud Volumes ONTAP 모범 사례

Trident 위한 ONTAP 및 Cloud Volumes ONTAP 구성에 대한 모범 사례를 알아보세요.

다음 권장 사항은 Trident 에서 동적으로 프로비저닝되는 볼륨을 사용하는 컨테이너화된 워크로드에 대해 ONTAP 구성하기 위한 지침입니다. 각 항목은 귀하의 환경에 적합한지 여부를 고려하여 평가해야 합니다.

Trident 에 전용된 **SVM**을 사용하세요

SVM(스토리지 가상 머신)은 ONTAP 시스템의 테넌트 간에 격리 및 관리적 분리를 제공합니다. SVM을 애플리케이션에 전용하면 권한을 위임하고 리소스 소비를 제한하는 모범 사례를 적용할 수 있습니다.

SVM 관리에는 여러 가지 옵션이 있습니다.

- 백엔드 구성에서 클러스터 관리 인터페이스를 적절한 자격 증명과 함께 제공하고 SVM 이름을 지정합니다.
- ONTAP 시스템 관리자나 CLI를 사용하여 SVM에 대한 전용 관리 인터페이스를 만듭니다.
- NFS 데이터 인터페이스와 관리 역할을 공유합니다.

각각의 경우, 인터페이스는 DNS에 있어야 하며, Trident 구성할 때 DNS 이름을 사용해야 합니다. 이를 통해 네트워크 ID 보존을 사용하지 않고도 SVM-DR과 같은 일부 DR 시나리오를 원활하게 진행할 수 있습니다.

SVM에 대한 전용 관리 LIF와 공유 관리 LIF 중 어느 것을 선호하느냐는 없습니다. 그러나 선택한 접근 방식에 맞게 네트워크 보안 정책이 일치하는지 확인해야 합니다. 그럼에도 불구하고 관리 LIF는 최대 유연성을 촉진하기 위해 DNS를 통해 액세스할 수 있어야 합니다. "SVM-DR" Trident 와 함께 사용할 수 있습니다.

최대 볼륨 수 제한

ONTAP 스토리지 시스템에는 최대 볼륨 수가 있으며, 이는 소프트웨어 버전과 하드웨어 플랫폼에 따라 다릅니다. 참조하다 "NetApp Hardware Universe" 정확한 한도를 확인하려면 해당 플랫폼과 ONTAP 버전을 확인하세요. 볼륨 수가 소진되면 Trident 뿐만 아니라 모든 스토리지 요청에 대한 프로비저닝 작업이 실패합니다.

트라이던트의 `ontap-nas` 그리고 `ontap-san` 드라이버는 생성된 각 Kubernetes 영구 볼륨(PV)에 대해 FlexVolume을 프로비저닝합니다. 그만큼 `ontap-nas-economy` 드라이버는 200개의 PV마다 약 1개의 FlexVolume을 생성합니다(50~300 사이로 구성 가능). 그만큼 `ontap-san-economy` 드라이버는 100개의 PV마다 약 1개의 FlexVolume을 생성합니다(50~200 사이로 구성 가능). Trident 스토리지 시스템의 사용 가능한 모든 볼륨을 소모하지 못하도록 하려면 SVM에 제한을 설정해야 합니다. 명령줄에서 이 작업을 수행할 수 있습니다.

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

에 대한 가치 `max-volumes` 사용자 환경에 따라 여러 가지 기준에 따라 다릅니다.

- ONTAP 클러스터에 있는 기존 볼륨의 수
- 다른 애플리케이션을 위해 Trident 외부에서 프로비저닝할 것으로 예상되는 볼륨 수
- Kubernetes 애플리케이션에서 사용될 것으로 예상되는 영구 볼륨 수

그만큼 `max-volumes` 값은 개별 ONTAP 노드가 아닌 ONTAP 클러스터의 모든 노드에 프로비저닝된 총 볼륨입니다. 결과적으로 ONTAP 클러스터 노드가 다른 노드보다 훨씬 더 많거나 적은 Trident 프로비저닝 볼륨을 갖는 경우가 발생할 수 있습니다.

예를 들어, 2노드 ONTAP 클러스터는 최대 2000개의 FlexVol 볼륨을 호스팅할 수 있습니다. 최대 볼륨 수를 1250으로 설정하는 것은 매우 타당한 것으로 보입니다. 그러나 만약에만 "집계" 한 노드에서 SVM에 할당되거나 한 노드에서 할당된 집계를 프로비저닝할 수 없는 경우(예: 용량 문제로 인해), 다른 노드가 모든 Trident 프로비저닝 볼륨의 대상이 됩니다. 이는 해당 노드의 볼륨 제한에 도달했을 수 있음을 의미합니다. `max-volumes` 값에 도달하면 Trident 와 해당 노드를 사용하는 다른 볼륨 작업에 영향을 미칩니다. 클러스터의 각 노드에서 집계된 데이터가 **Trident** 에서 사용하는 **SVM**에 동일한 개수로 할당되도록 하면 이러한 상황을 피할 수 있습니다**.**

볼륨 복제

NetApp Trident 다음을 사용할 때 볼륨 복제를 지원합니다. `ontap-nas` , `ontap-san` , `solidfire-san` , 그리고 `gcp-cvs` 스토리지 드라이버. 사용 시 `ontap-nas-flexgroup` 또는 `ontap-nas-economy` 드라이버, 복제가 지원되지 않습니다. 기존 볼륨에서 새 볼륨을 생성하면 새로운 스냅샷이 생성됩니다.

> 다른 StorageClass와 연결된 PVC를 복제하지 마세요. 호환성을 보장하고 예상치 못한 동작을 방지하려면 동일한 StorageClass 내에서 복제 작업을 수행합니다.

Trident 에서 생성된 볼륨의 최대 크기 제한

Trident 에서 생성할 수 있는 볼륨의 최대 크기를 구성하려면 다음을 사용하십시오. `limitVolumeSize` 귀하의 매개변수 `backend.json` 정의.

스토리지 어레이에서 볼륨 크기를 제어하는 것 외에도 Kubernetes 기능도 활용해야 합니다.

Trident 에서 생성된 **FlexVol**의 최대 크기 제한

ontap-san-economy 및 ontap-nas-economy 드라이버의 풀로 사용되는 FlexVol의 최대 크기를 구성하려면 다음을 사용하세요. `limitVolumePoolSize` 귀하의 매개변수 `backend.json` 정의.

양방향 **CHAP**를 사용하도록 **Trident** 구성

백엔드 정의에서 CHAP 초기자 및 대상 사용자 이름과 비밀번호를 지정하고 Trident SVM에서 CHAP를 활성화하도록 할 수 있습니다. 를 사용하여 `useCHAP` 백엔드 구성의 매개변수를 통해 Trident CHAP를 사용하여 ONTAP 백엔드에 대한 iSCSI 연결을 인증합니다.

SVM QoS 정책 생성 및 사용

SVM에 적용된 ONTAP QoS 정책을 활용하면 Trident 프로비저닝 볼륨에서 사용할 수 있는 IOPS 수가 제한됩니다. 이것은 도움이 됩니다 "괴롭힘을 예방하다" 또는 Trident SVM 외부의 작업 부하에 영향을 미치는 제어 불가능한 컨테이너입니다.

몇 단계만으로 SVM에 대한 QoS 정책을 만들 수 있습니다. 가장 정확한 정보는 해당 ONTAP 버전의 설명서를 참조하세요. 아래 예에서는 SVM에서 사용할 수 있는 총 IOPS를 5000으로 제한하는 QoS 정책을 만듭니다.

```
# 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>
```

또한 ONTAP 버전에서 지원하는 경우 컨테이너화된 워크로드에 대한 처리량을 보장하기 위해 QoS 최소값을 사용하는 것을 고려할 수 있습니다. 적응형 QoS는 SVM 수준 정책과 호환되지 않습니다.

컨테이너화된 워크로드에 전담된 IOPS 수는 여러 측면에 따라 달라집니다. 여기에는 다음이 포함됩니다.

- 스토리지 어레이를 사용하는 다른 워크로드. Kubernetes 배포와 관련이 없는 다른 워크로드가 스토리지 리소스를 활용하는 경우, 해당 워크로드가 실수로 부정적인 영향을 받지 않도록 주의해야 합니다.
- 컨테이너에서 실행되는 예상 작업 부하. 높은 IOPS 요구 사항이 있는 워크로드가 컨테이너에서 실행되는 경우, 낮은 QoS 정책은 나쁜 사용자 경험을 초래합니다.

SVM 수준에서 할당된 QoS 정책은 SVM에 프로비저닝된 모든 볼륨이 동일한 IOPS 풀을 공유하게 된다는 점을 기억하는 것이 중요합니다. 컨테이너화된 애플리케이션 중 하나 또는 소수가 높은 IOPS 요구 사항을 갖는 경우 다른 컨테이너화된 워크로드에 방해가 될 수 있습니다. 이 경우 외부 자동화를 사용하여 볼륨별 QoS 정책을 할당하는 것을 고려해 볼 수 있습니다.

ONTAP 버전이 9.8 이전인 경우에만 QoS 정책 그룹을 SVM에 할당해야 합니다.

Trident 에 대한 **QoS** 정책 그룹 생성

서비스 품질(QoS)은 중요한 작업 부하의 성능이 경쟁 작업 부하로 인해 저하되지 않도록 보장합니다. ONTAP QoS 정책 그룹은 볼륨에 대한 QoS 옵션을 제공하고 사용자가 하나 이상의 작업 부하에 대한 처리량 상한을 정의할 수 있도록 합니다. QoS에 대한 자세한 내용은 다음을 참조하세요. "QoS로 처리량 보장" . 백엔드나 스토리지 풀에서 QoS 정책 그룹을 지정할 수 있으며, 해당 정책 그룹은 해당 풀이나 백엔드에서 생성된 각 볼륨에 적용됩니다.

ONTAP 에는 기존 QoS 정책 그룹과 적응형 QoS 정책 그룹 두 가지가 있습니다. 기존 정책 그룹은 IOPS에서 일정한 최대(또는 이후 버전에서는 최소) 처리량을 제공합니다. 적응형 QoS는 작업 부하 크기에 맞게 처리량을 자동으로 조정하여 작업 부하 크기가 변해도 IOPS 대 TB|GB 비율을 유지합니다. 이는 대규모 배포에서 수백 또는 수천 개의 작업 부하를 관리할 때 상당한 이점을 제공합니다.

QoS 정책 그룹을 생성할 때 다음 사항을 고려하세요.

- 당신은 설정해야합니다 `qosPolicy` 키에 `defaults` 백엔드 구성의 블록. 다음 백엔드 구성 예를 참조하세요.

```
---
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
```

- 정책 그룹에서 지정한 대로 각 볼륨이 전체 처리량을 얻도록 볼륨별로 정책 그룹을 적용해야 합니다. 공유 정책 그룹은 지원되지 않습니다.

QoS 정책 그룹에 대한 자세한 내용은 다음을 참조하세요. "ONTAP 명령 참조" .

Kubernetes 클러스터 멤버에 대한 스토리지 리소스 액세스 제한

Trident 가 생성한 NFS 볼륨, iSCSI LUN 및 FC LUN에 대한 액세스를 제한하는 것은 Kubernetes 배포에 대한 보안 태세의 중요한 구성 요소입니다. 이렇게 하면 Kubernetes 클러스터에 속하지 않은 호스트가 볼륨에 액세스하여 예기치 않게 데이터를 수정할 가능성이 방지됩니다.

네임스페이스가 Kubernetes 리소스의 논리적 경계라는 것을 이해하는 것이 중요합니다. 동일한 네임스페이스에 있는 리소스는 공유할 수 있다고 가정하지만, 중요한 점은 네임스페이스 간 기능이 없다는 것입니다. 즉, PV는 글로벌 객체이지만 PVC에 바인딩되면 동일한 네임스페이스에 있는 Pod에서만 액세스할 수 있습니다. 적절한 경우 네임스페이스를 사용하여 분리하는 것이 중요합니다**.**

Kubernetes 컨텍스트에서 데이터 보안과 관련하여 대부분 조직이 가장 우려하는 점은 컨테이너의 프로세스가 호스트에 마운트된 저장소에 액세스할 수 있지만 해당 저장소가 컨테이너용으로 의도되지 않았다는 것입니다. "네임스페이스" 이러한 유형의 손상을 방지하기 위해 설계되었습니다. 하지만 예외가 하나 있습니다. 특권 컨테이너입니다.

특권 컨테이너는 일반적인 컨테이너보다 훨씬 더 많은 호스트 수준 권한으로 실행되는 컨테이너입니다. 이러한 기능은 기본적으로 거부되지 않으므로 다음을 사용하여 기능을 비활성화해야 합니다. "포드 보안 정책" .

Kubernetes와 외부 호스트 모두에서 액세스가 필요한 볼륨의 경우, PV는 관리자가 도입하고 Trident 가 관리하지 않는 기존 방식으로 스토리지를 관리해야 합니다. 이렇게 하면 Kubernetes와 외부 호스트가 모두 연결이 끊기고 더 이상 볼륨을 사용하지 않을 때만 스토리지 볼륨이 파기됩니다. 또한 사용자 정의 내보내기 정책을 적용하면 Kubernetes 클러스터 노드와 Kubernetes 클러스터 외부의 대상 서버에서 액세스할 수 있습니다.

전용 인프라 노드(예: OpenShift)나 사용자 애플리케이션을 예약할 수 없는 다른 노드가 있는 배포의 경우, 별도의 내보내기 정책을 사용하여 스토리지 리소스에 대한 액세스를 추가로 제한해야 합니다. 여기에는 인프라 노드에 배포된 서비스(예: OpenShift Metrics 및 Logging 서비스)와 비인프라 노드에 배포된 표준 애플리케이션에 대한 내보내기 정책을 만드는 작업이 포함됩니다.

전담 수출 정책을 사용하세요

Kubernetes 클러스터에 있는 노드에만 액세스를 허용하는 각 백엔드에 대한 내보내기 정책이 있는지 확인해야 합니다. Trident 자동으로 수출 정책을 만들고 관리할 수 있습니다. 이런 방식으로 Trident Kubernetes 클러스터의 노드에 프로비저닝하는 볼륨에 대한 액세스를 제한하고 노드 추가/삭제를 간소화합니다.

또는 수동으로 내보내기 정책을 만들고 각 노드 액세스 요청을 처리하는 하나 이상의 내보내기 규칙으로 채울 수도 있습니다.

- 사용하다 `vserver export-policy create` ONTAP CLI 명령을 사용하여 내보내기 정책을 생성합니다.
- 다음을 사용하여 내보내기 정책에 규칙을 추가합니다. `vserver export-policy rule create` ONTAP CLI 명령.

이러한 명령을 실행하면 어떤 Kubernetes 노드가 데이터에 액세스할 수 있는지 제한할 수 있습니다.

장애를 입히다 `showmount` **SVM** 응용 프로그램을 위해

그만큼 `showmount` 이 기능을 사용하면 NFS 클라이언트가 SVM에 사용 가능한 NFS 내보내기 목록을 쿼리할 수 있습니다. Kubernetes 클러스터에 배포된 Pod는 다음을 발행할 수 있습니다. `showmount -e` 명령을 내리고, 접근할

수 없는 탈것을 포함하여 사용 가능한 탈것의 목록을 받습니다. 이것 자체는 보안을 위협하는 것은 아니지만 불필요한 정보를 제공하여 권한이 없는 사용자가 NFS 내보내기에 연결하는 데 도움이 될 수 있습니다.

비활성화해야 합니다 `showmount` SVM 수준 ONTAP CLI 명령을 사용하여:

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

SolidFire 모범 사례

Trident 위한 SolidFire 스토리지 구성에 대한 모범 사례를 알아보세요.

Solidfire 계정 생성

각 SolidFire 계정은 고유한 볼륨 소유자를 나타내며 고유한 CHAP(Challenge-Handshake 인증 프로토콜) 자격 증명 세트를 받습니다. 계정 이름과 관련 CHAP 자격 증명을 사용하거나 볼륨 액세스 그룹을 통해 계정에 할당된 볼륨에 액세스할 수 있습니다. 한 계정에는 최대 2,000개의 볼륨이 할당될 수 있지만, 볼륨은 한 계정에만 속할 수 있습니다.

QoS 정책 생성

여러 볼륨에 적용할 수 있는 표준화된 서비스 품질 설정을 만들고 저장하려면 SolidFire 서비스 품질(QoS) 정책을 사용하세요.

볼륨별로 QoS 매개변수를 설정할 수 있습니다. 각 볼륨의 성능은 QoS를 정의하는 세 가지 구성 가능한 매개변수(최소 IOPS, 최대 IOPS, 버스트 IOPS)를 설정하여 보장할 수 있습니다.

4Kb 블록 크기에 대한 최소, 최대 및 버스트 IOPS 값은 다음과 같습니다.

IOPS 매개변수	정의	최소값	기본값	최대값(4Kb)
최소 IOPS	볼륨에 대해 보장된 성능 수준입니다.	50	50	15000
최대 IOPS	성능은 이 한도를 초과하지 않습니다.	50	15000	200,000
버스트 IOPS	짧은 버스트 시나리오에서 허용되는 최대 IOPS입니다.	50	15000	200,000

최대 IOPS와 버스트 IOPS는 최대 200,000까지 설정할 수 있지만, 볼륨의 실제 최대 성능은 클러스터 사용량과 노드당 성능에 따라 제한됩니다.

블록 크기와 대역폭은 IOPS 수에 직접적인 영향을 미칩니다. 블록 크기가 증가함에 따라 시스템은 더 큰 블록 크기를 처리하는 데 필요한 수준으로 대역폭을 늘립니다. 대역폭이 증가함에 따라 시스템이 달성할 수 있는 IOPS 수는 감소합니다. 참조하다 "SolidFire 서비스 품질" QoS 및 성능에 대한 자세한 내용은 다음을 참조하세요.

SolidFire 인증

Element는 CHAP 및 VAG(볼륨 액세스 그룹)라는 두 가지 인증 방법을 지원합니다. CHAP는 CHAP 프로토콜을 사용하여 백엔드에 호스트를 인증합니다. 볼륨 액세스 그룹은 프로비저닝하는 볼륨에 대한 액세스를 제어합니다. NetApp 인증에 CHAP를 사용할 것을 권장합니다. CHAP는 더 간단하고 확장 제한이 없기 때문입니다.

> 향상된 CSI 프로비저너가 탑재된 Trident CHAP 인증 사용을 지원합니다. VAG는 기존의 비 CSI 작동 모드에서만 사용해야 합니다.

CHAP 인증(개시자가 의도된 볼륨 사용자인지 확인하는 것)은 계정 기반 액세스 제어를 통해서만 지원됩니다. 인증에 CHAP를 사용하는 경우 단방향 CHAP와 양방향 CHAP의 두 가지 옵션을 사용할 수 있습니다. 단방향 CHAP는 SolidFire 계정 이름과 개시자 비밀번호를 사용하여 볼륨 액세스를 인증합니다. 양방향 CHAP 옵션은 볼륨이 계정 이름과 개시자 비밀번호를 통해 호스트를 인증하고, 그런 다음 호스트가 계정 이름과 대상 비밀번호를 통해 볼륨을 인증하기 때문에 볼륨을 인증하는 가장 안전한 방법을 제공합니다.

하지만 CHAP를 활성화할 수 없고 VAG가 필요한 경우 액세스 그룹을 만들고 호스트 이니시에이터와 볼륨을 액세스 그룹에 추가합니다. 액세스 그룹에 추가하는 각 IQN은 CHAP 인증 여부와 관계없이 그룹의 각 볼륨에 액세스할 수 있습니다. iSCSI 이니시에이터가 CHAP 인증을 사용하도록 구성된 경우 계정 기반 액세스 제어가 사용됩니다. iSCSI 이니시에이터가 CHAP 인증을 사용하도록 구성되지 않은 경우 볼륨 액세스 그룹 액세스 제어가 사용됩니다.

더 많은 정보는 어디에서 찾을 수 있나요?

다음은 모범 사례 문서 중 일부입니다. 검색 "NetApp 라이브러리" 최신 버전을 보려면 여기를 클릭하세요.

- ONTAP*
- "NFS 모범 사례 및 구현 가이드"
- "SAN 관리"(iSCSI용)
- "RHEL용 iSCSI Express 구성"

엘리먼트 소프트웨어

- "Linux용 SolidFire 구성"
- NetApp HCI*
- "NetApp HCI 배포 전제 조건"
- "NetApp 배포 엔진에 액세스"

응용 프로그램 모범 사례 정보

- "ONTAP 에서 MySQL을 위한 모범 사례"
- "SolidFire 에서 MySQL을 위한 모범 사례"
- "NetApp SolidFire 및 Cassandra"
- "SolidFire 에 대한 Oracle 모범 사례"
- "SolidFire 에서의 PostgreSQL 모범 사례"

모든 애플리케이션에 특정 지침이 있는 것은 아니므로 NetApp 팀과 협력하여 사용하는 것이 중요합니다. "NetApp 라이브러리" 가장 최신 문서를 찾으려면.

Trident 통합

Trident 통합하려면 드라이버 선택 및 배포, 스토리지 클래스 설계, 가상 풀 설계, 스토리지 프로비저닝에 미치는 PVC(영구 볼륨 클레임)의 영향, 볼륨 작업, Trident 사용한 OpenShift 서비스 배포 등의 설계 및 아키텍처 요소를 통합해야 합니다.

드라이버 선택 및 배치

스토리지 시스템에 맞는 백엔드 드라이버를 선택하고 배포합니다.

ONTAP 백엔드 드라이버

ONTAP 백엔드 드라이버는 사용되는 프로토콜과 스토리지 시스템에서 볼륨이 프로비저닝되는 방식에 따라 구분됩니다. 따라서 어떤 드라이버를 배치할지 결정할 때 신중하게 고려하세요.

더 높은 수준에서, 애플리케이션에 공유 스토리지(동일한 PVC에 액세스하는 여러 포드)가 필요한 구성 요소가 있는 경우 NAS 기반 드라이버가 기본 선택이 되고, 블록 기반 iSCSI 드라이버는 공유되지 않는 스토리지의 요구 사항을 충족합니다. 애플리케이션의 요구 사항과 스토리지 및 인프라 팀의 편의성 수준에 따라 프로토콜을 선택하세요. 일반적으로 대부분의 애플리케이션에서는 두 가지 사이에 큰 차이가 없으므로, 공유 스토리지(두 개 이상의 포드에 동시에 액세스해야 하는 경우)가 필요한지 여부에 따라 결정이 내려지는 경우가 많습니다.

사용 가능한 ONTAP 백엔드 드라이버는 다음과 같습니다.

- `ontap-nas`: 프로비저닝된 각 PV는 전체 ONTAP FlexVolume입니다.
- `ontap-nas-economy`: 프로비저닝된 각 PV는 qtree이며, FlexVolume당 구성 가능한 qtree 수가 있습니다 (기본값은 200).
- `ontap-nas-flexgroup`: 각 PV는 전체 ONTAP FlexGroup 으로 프로비저닝되고, SVM에 할당된 모든 집계가 사용됩니다.
- `ontap-san`: 프로비저닝된 각 PV는 자체 FlexVolume 내의 LUN입니다.
- `ontap-san-economy`: 프로비저닝된 각 PV는 LUN이며, FlexVolume당 LUN 수는 구성 가능합니다(기본값은 100).

세 가지 NAS 드라이버 중에서 선택하면 애플리케이션에서 사용할 수 있는 기능에 몇 가지 영향이 있습니다.

아래 표에서 모든 기능이 Trident 통해 공개되는 것은 아니라는 점에 유의하세요. 해당 기능이 필요한 경우 스토리지 관리자가 프로비저닝 후에 일부 기능을 적용해야 합니다. 상위 첨자 각주는 기능과 드라이버별 기능을 구분합니다.

ONTAP NAS 드라이버	스냅샷	클론	동적 수출 정책	다중 부착	서비스 품질	크기 조정	복제
`ontap-nas`	예	예	예각주:5[]	예	예각주:1[]	예	예각주:1[]
`ontap-nas-economy`	NO각주:3[]	NO각주:3[]	예각주:5[]	예	NO각주:3[]	예	NO각주:3[]
`ontap-nas-flexgroup`	예각주:1[]	아니요	예각주:5[]	예	예각주:1[]	예	예각주:1[]

Trident ONTAP 용 SAN 드라이버 2개를 제공하며, 해당 드라이버의 기능은 아래와 같습니다.

ONTAP SAN 드라이버	스냅샷	클론	다중 부착	양방향 **CHAP**	서비스 품질	크기 조정	복제
`ontap-san`	예	예	예각주:4[]	예	예각주:1[]	예	예각주:1[]
`ontap-san-economy`	예	예	예각주:4[]	예	NO각주:3[]	예	NO각주:3[]

> 위 표에 대한 각주: 예각주:1[]: Trident 에서 관리하지 않음 예각주:2[]: Trident 에서 관리하지만 PV 세분화되지 않음 아니요각주:3[]: Trident 에서 관리하지 않고 PV 세분화되지 않음 예각주:4[]: 원시 블록 볼륨에 대해 지원됨 예각주:5[]: Trident 에서 지원됨

PV 세분화가 아닌 기능은 전체 FlexVolume에 적용되며 모든 PV(즉, 공유 FlexVol의 qtree 또는 LUN)는 공통 일정을 공유합니다.

위의 표에서 볼 수 있듯이, 다음 기능 간의 많은 부분이 `ontap-nas` 그리고 `ontap-nas-economy` 동일합니다. 그러나, `ontap-nas-economy` 드라이버는 PV 단위로 일정을 제어하는 기능을 제한하며, 이는 특히 재해 복구 및 백업 계획에 영향을 미칠 수 있습니다. ONTAP 스토리지에서 PVC 복제 기능을 활용하려는 개발 팀의 경우 이는 다음을 사용할 때만 가능합니다. `ontap-nas` , `ontap-san` 또는 `ontap-san-economy` 운전자.

> 그만큼 `solidfire-san` 드라이버는 PVC를 복제할 수도 있습니다.

Cloud Volumes ONTAP 백엔드 드라이버

Cloud Volumes ONTAP 파일 공유 및 NAS 및 SAN 프로토콜(NFS, SMB/CIFS, iSCSI)을 제공하는 블록 수준 스토리지를 포함하여 다양한 사용 사례에 대한 엔터프라이즈급 스토리지 기능과 함께 데이터 제어 기능을 제공합니다. Cloud Volume ONTAP 에 호환되는 드라이버는 다음과 같습니다. `ontap-nas` , `ontap-nas-economy` , `ontap-san` 그리고 `ontap-san-economy` . 이러한 기능은 Azure용 Cloud Volume ONTAP , GCP용 Cloud Volume ONTAP 에 적용됩니다.

Amazon FSx for ONTAP 백엔드 드라이버

Amazon FSx for NetApp ONTAP 사용하면 AWS에 데이터를 저장함으로써 얻는 단순성, 민첩성, 보안 및 확장성의 이점을 누리면서 익숙한 NetApp 기능, 성능 및 관리 역량을 활용할 수 있습니다. FSx for ONTAP 다양한 ONTAP 파일 시스템 기능과 관리 API를 지원합니다. Cloud Volume ONTAP 에 호환되는 드라이버는 다음과 같습니다. `ontap-nas` , `ontap-nas-economy` , `ontap-nas-flexgroup` , `ontap-san` 그리고 `ontap-san-economy` .

NetApp HCI/ SolidFire 백엔드 드라이버

그만큼 `solidfire-san` NetApp HCI/ SolidFire 플랫폼과 함께 사용되는 드라이버는 관리자가 QoS 제한에 따라 Trident 에 대한 Element 백엔드를 구성하는 데 도움이 됩니다. Trident 에서 프로비저닝한 볼륨에 대한 특정 QoS 제한을 설정하도록 백엔드를 설계하려면 다음을 사용하십시오. `type` 백엔드 파일의 매개변수. 관리자는 또한 다음을 사용하여 저장소에서 생성될 수 있는 볼륨 크기를 제한할 수 있습니다. `limitVolumeSize` 매개변수. 현재 볼륨 크기 조정 및 볼륨 복제와 같은 Element 스토리지 기능은 지원되지 않습니다. `solidfire-san` 운전사. 이러한 작업은 Element Software 웹 UI를 통해 수동으로 수행해야 합니다.

SolidFire 드라이버	스냅샷	클론	다중 부착	녀석	서비스 품질	크기 조정	복제
`solidfire-san`	예	예	예각주:2[]	예	예	예	예각주:1[]

> 각주: 예각주:1[]: Trident 에서 관리되지 않음 예각주:2[]: 원시 블록 볼륨에 대해 지원됨

Azure NetApp Files 백엔드 드라이버

Trident 다음을 사용합니다. `azure-netapp-files` 관리하는 드라이버"Azure NetApp Files" 서비스.

이 드라이버에 대한 자세한 정보와 이를 구성하는 방법은 다음에서 확인할 수 있습니다."Azure NetApp Files 대한 Trident 백엔드 구성" .

Azure NetApp Files 드라이버	스냅샷	클론	다중 부착	서비스 품질	확장하다	복제
`azure-netapp-files`	예	예	예	예	예	예각주:1[]

각주: 예각주:1[]: Trident 에서 관리하지 않음

Google Cloud 백엔드 드라이버의 Cloud Volumes Service

Trident 다음을 사용합니다. `gcp-cvs` Google Cloud의 Cloud Volumes Service 에 연결하는 드라이버입니다.

그만큼 `gcp-cvs` 드라이버는 가상 풀을 사용하여 백엔드를 추상화하고 Trident 볼륨 배치를 결정할 수 있도록 합니다. 관리자는 가상 풀을 정의합니다. `backend.json` 파일. 저장소 클래스는 선택기를 사용하여 레이블로 가상 풀을 식별합니다.

- 백엔드에 가상 풀이 정의된 경우 Trident 해당 가상 풀이 제한된 Google Cloud 스토리지 풀에 볼륨을 생성하려고 시도합니다.
- 백엔드에 가상 풀이 정의되어 있지 않으면 Trident 해당 지역의 사용 가능한 스토리지 풀 중에서 Google Cloud 스토리지 풀을 선택합니다.

Trident 에서 Google Cloud 백엔드를 구성하려면 다음을 지정해야 합니다. `projectNumber` , `apiRegion` , 그리고 `apiKey` 백엔드 파일에서. 프로젝트 번호는 Google Cloud 콘솔에서 찾을 수 있습니다. API 키는 Google Cloud에서 Cloud Volumes Service 에 대한 API 액세스를 설정할 때 생성한 서비스 계정 개인 키 파일에서 가져옵니다.

Google Cloud 서비스 유형 및 서비스 수준의 Cloud Volumes Service 에 대한 자세한 내용은 다음을 참조하세요 ."GCP용 CVS에 대한 Trident 지원에 대해 알아보세요" .

Google Cloud 드라이버용 **Cloud Volumes Service**	스냅샷	클론	다중 부착	서비스 품질	확장하다	복제
`gcp-cvs`	예	예	예	예	예	CVS-Performance 서비스 유형에서만 사용 가능합니다.

복제 노트

- 복제는 Trident 에서 관리되지 않습니다.
- 복제본은 소스 볼륨과 동일한 스토리지 풀에 생성됩니다.

스토리지 클래스 디자인

Kubernetes 스토리지 클래스 객체를 생성하려면 개별 스토리지 클래스를 구성하고 적용해야 합니다. 이 섹션에서는 애플리케이션에 맞는 스토리지 클래스를 설계하는 방법을 설명합니다.

특정 백엔드 활용

필터링은 특정 스토리지 클래스 개체 내에서 사용하여 해당 특정 스토리지 클래스와 함께 사용할 스토리지 풀 또는 풀 세트를 결정할 수 있습니다. 스토리지 클래스에는 세 가지 필터 세트를 설정할 수 있습니다. `storagePools` , `additionalStoragePools` , 및/또는 `excludeStoragePools` .

그만큼 `storagePools` 매개변수는 지정된 속성과 일치하는 풀 세트로 저장소를 제한하는 데 도움이 됩니다. 그만큼 `additionalStoragePools` 매개변수는 Trident 가 프로비저닝에 사용하는 풀 세트와 속성에 의해 선택된 풀 세트를 확장하는 데 사용됩니다. `storagePools` 매개변수. 두 매개변수 중 하나만 사용하거나 두 매개변수를 함께 사용하여 적절한 스토리지 풀 세트가 선택되었는지 확인할 수 있습니다.

그만큼 `excludeStoragePools` 매개변수는 속성과 일치하는 풀의 나열된 세트를 구체적으로 제외하는 데 사용됩니다.

QoS 정책 에뮬레이션

서비스 품질 정책을 에뮬레이트하기 위해 스토리지 클래스를 설계하려면 다음을 사용하여 스토리지 클래스를 만듭니다. `media` 속성으로 `hdd` 또는 `ssd` . 에 근거하여 `media` 저장 클래스에 언급된 속성에 대해 Trident 적절한 백엔드를 선택합니다. `hdd` 또는 `ssd` 미디어 속성과 일치하는 집계를 수행한 다음 볼륨 프로비저닝을 특정 집계로 지시합니다. 따라서 PREMIUM 스토리지 클래스를 생성할 수 있습니다. `media` 속성 집합으로 `ssd` 이는 PREMIUM QoS 정책으로 분류될 수 있습니다. 또 다른 저장 클래스 STANDARD를 만들면 미디어 속성이 `hdd`로 설정되고, 이는 STANDARD QoS 정책으로 분류될 수 있습니다. 저장소 클래스에서 ``IOPS'' 속성을 사용하여 프로비저닝을 QoS 정책으로 정의할 수 있는 Element 어플라이언스로 리디렉션할 수도 있습니다.

특정 기능에 기반한 백엔드 활용

스토리지 클래스는 씬 및 씩 프로비저닝, 스냅샷, 복제, 암호화 등의 기능이 활성화된 특정 백엔드에서 볼륨 프로비저닝을 지시하도록 설계될 수 있습니다. 사용할 저장소를 지정하려면 필요한 기능이 활성화된 적절한 백엔드를 지정하는 저장소 클래스를 만듭니다.

가상 풀

모든 Trident 백엔드에서 가상 풀을 사용할 수 있습니다. Trident 제공하는 모든 드라이버를 사용하여 모든 백엔드에 대한 가상 풀을 정의할 수 있습니다.

가상 풀을 사용하면 관리자가 백엔드에 대한 추상화 수준을 생성할 수 있으며, 이는 스토리지 클래스를 통해 참조할 수 있어 백엔드에 볼륨을 보다 유연하게 배치하고 효율적으로 배치할 수 있습니다. 동일한 서비스 클래스로 서로 다른 백엔드를 정의할 수 있습니다. 게다가 동일한 백엔드에 서로 다른 특성을 가진 여러 개의 스토리지 풀을 만들 수도 있습니다. 스토리지 클래스가 특정 레이블이 있는 선택기로 구성된 경우 Trident 모든 선택기 레이블과 일치하는 백엔드를 선택하여 볼륨을 배치합니다. 스토리지 클래스 선택기 레이블이 여러 스토리지 풀과 일치하는 경우 Trident 볼륨을 프로비저닝할 스토리지 풀을 하나 선택합니다.

가상 풀 디자인

백엔드를 생성할 때 일반적으로 매개변수 집합을 지정할 수 있습니다. 이전에는 관리자가 동일한 스토리지 자격 증명과 다른 매개변수 집합을 사용하여 다른 백엔드를 생성하는 것이 불가능했습니다. 가상 풀이 도입되면서 이 문제가 해결되었습니다. 가상 풀은 백엔드와 Kubernetes 스토리지 클래스 사이에 도입된 레벨 추상화로, 관리자가 백엔드에

관계없이 Kubernetes 스토리지 클래스를 통해 선택기로 참조할 수 있는 레이블과 함께 매개변수를 정의할 수 있도록 합니다. 가상 풀은 Trident 를 통해 지원되는 모든 NetApp 백엔드에 대해 정의할 수 있습니다. 해당 목록에는 SolidFire/NetApp HCI, ONTAP, GCP의 Cloud Volumes Service 및 Azure NetApp Files 포함됩니다.

> 가상 풀을 정의할 때 백엔드 정의에서 기존 가상 풀의 순서를 재정렬하지 않는 것이 좋습니다. 기존 가상 풀의 속성을 편집/수정하지 않고 대신 새 가상 풀을 정의하는 것이 좋습니다.

다양한 서비스 수준/**QoS** 에뮬레이션

서비스 클래스를 에뮬레이션하기 위해 가상 풀을 설계하는 것이 가능합니다. Azure NetApp Files 용 Cloud Volume Service의 가상 풀 구현을 사용하여 다양한 서비스 클래스를 설정하는 방법을 살펴보겠습니다. 다양한 성능 수준을 나타내는 여러 레이블로 Azure NetApp Files 백엔드를 구성합니다. 세트 `servicelevel` 각 라벨 아래에 필요한 다른 측면을 추가하고, 적절한 성능 수준에 맞춰 측면을 조정합니다. 이제 다양한 가상 풀에 매핑되는 다양한 Kubernetes 스토리지 클래스를 만듭니다. 를 사용하여 `parameters.selector` 필드에서 각 StorageClass는 볼륨을 호스팅하는 데 사용할 수 있는 가상 풀을 호출합니다.

특정 측면 집합 할당

특정 측면을 갖춘 여러 개의 가상 풀을 단일 스토리지 백엔드에서 설계할 수 있습니다. 이를 위해 백엔드를 여러 개의 레이블로 구성하고 각 레이블 아래에 필요한 측면을 설정합니다. 이제 다음을 사용하여 다양한 Kubernetes 스토리지 클래스를 만듭니다. `parameters.selector` 다양한 가상 풀에 매핑되는 필드입니다. 백엔드에 프로비저닝되는 볼륨에는 선택한 가상 풀에 정의된 측면이 있습니다.

저장 용량에 영향을 미치는 **PVC** 특성

요청된 스토리지 클래스를 넘어서는 일부 매개변수는 PVC를 생성할 때 Trident 프로비저닝 결정 프로세스에 영향을 미칠 수 있습니다.

접근 모드

PVC를 통해 저장소를 요청할 때 필수 필드 중 하나는 액세스 모드입니다. 원하는 모드는 저장 요청을 호스팅하기 위해 선택된 백엔드에 영향을 미칠 수 있습니다.

Trident 다음 매트릭스에 따라 지정된 액세스 방법과 사용된 저장 프로토콜을 일치시키려고 시도합니다. 이는 기본 저장 플랫폼과 무관합니다.

	한 번 읽기/쓰기	읽기 전용 다수	읽기쓰기많음
iSCSI	예	예	네 (원시 블록)
NFS	예	예	예

NFS 백엔드가 구성되지 않은 Trident 배포에 ReadWriteMany PVC에 대한 요청을 제출하면 볼륨이 프로비저닝되지 않습니다. 이러한 이유로 요청자는 자신의 애플리케이션에 적합한 액세스 모드를 사용해야 합니다.

볼륨 작업

영구 볼륨 수정

두 가지 예외를 제외하고 영구 볼륨은 Kubernetes에서 변경할 수 없는 객체입니다. 회수 정책과 크기를 생성한 후에는 이를 수정할 수 있습니다. 하지만 이렇게 해도 볼륨의 일부 측면이 Kubernetes 외부에서 수정되는 것을 막을 수는 없습니다. 이는 특정 애플리케이션에 맞게 볼륨을 사용자 지정하거나, 용량이 실수로 소모되는 것을 방지하거나, 어떤

이유로든 볼륨을 다른 스토리지 컨트롤러로 옮기는 경우에 바람직할 수 있습니다.

> Kubernetes 인트리 프로비저너는 현재 NFS, iSCSI 또는 FC PV에 대한 볼륨 크기 조정 작업을 지원하지 않습니다. Trident NFS, iSCSI, FC 볼륨 확장을 모두 지원합니다.

PV의 연결 세부정보는 생성 후 수정할 수 없습니다.

주문형 볼륨 스냅샷 만들기

Trident CSI 프레임워크를 사용하여 주문형 볼륨 스냅샷 생성과 스냅샷에서 PVC 생성을 지원합니다. 스냅샷은 데이터의 특정 시점 사본을 유지 관리하는 편리한 방법을 제공하며 Kubernetes의 소스 PV와 독립적인 수명 주기를 갖습니다. 이러한 스냅샷은 PVC를 복제하는 데 사용할 수 있습니다.

스냅샷에서 볼륨 생성

Trident 볼륨 스냅샷에서 PersistentVolume을 생성하는 기능도 지원합니다. 이를 달성하려면 PersistentVolumeClaim을 생성하고 다음을 언급하기만 하면 됩니다. `datasource` 볼륨을 생성하는 데 필요한 스냅샷입니다. Trident 스냅샷에 있는 데이터로 볼륨을 생성하여 이 PVC를 처리합니다. 이 기능을 사용하면 여러 지역에 걸쳐 데이터를 복제하고, 테스트 환경을 만들고, 손상되거나 훼손된 운영 볼륨을 전체적으로 교체하거나, 특정 파일과 디렉터리를 검색하여 다른 연결된 볼륨으로 전송할 수 있습니다.

클러스터에서 볼륨 이동

스토리지 관리자는 ONTAP 클러스터의 집계와 컨트롤러 간에 볼륨을 스토리지 소비자에게 중단 없이 이동할 수 있습니다. 대상 집계가 Trident 사용하는 SVM에서 액세스할 수 있는 집계인 한, 이 작업은 Trident 나 Kubernetes 클러스터에 영향을 미치지 않습니다. 중요한 점은, 집계가 SVM에 새로 추가된 경우 백엔드를 Trident 에 다시 추가하여 새로 고쳐야 한다는 것입니다. 이렇게 하면 Trident SVM을 재인벤토리하여 새로운 집계가 인식되도록 합니다.

하지만 백엔드 간 볼륨 이동은 Trident 에서 자동으로 지원되지 않습니다. 여기에는 동일한 클러스터 내의 SVM 간, 클러스터 간 또는 다른 스토리지 플랫폼(해당 스토리지 시스템이 Trident 에 연결된 경우에도 해당)이 포함됩니다.

볼륨을 다른 위치로 복사하는 경우 볼륨 가져오기 기능을 사용하여 현재 볼륨을 Trident 로 가져올 수 있습니다.

볼륨 확장

Trident NFS, iSCSI, FC PV의 크기 조정을 지원합니다. 이를 통해 사용자는 Kubernetes 계층을 통해 볼륨 크기를 직접 조정할 수 있습니다. ONTAP, SolidFire/ NetApp HCI 및 Cloud Volumes Service 백엔드를 포함한 모든 주요 NetApp 스토리지 플랫폼에서 볼륨 확장이 가능합니다. 나중에 확장이 가능하도록 설정하세요. `allowVolumeExpansion` 에게 `true` 볼륨과 연결된 StorageClass에서. 영구 볼륨의 크기를 조정해야 할 때마다 다음을 편집합니다. `spec.resources.requests.storage` 영구 볼륨 클레임에 필요한 볼륨 크기에 대한 주석을 추가합니다. Trident 스토리지 클러스터의 볼륨 크기를 자동으로 조정합니다.

기존 볼륨을 **Kubernetes**로 가져오기

볼륨 가져오기는 기존 스토리지 볼륨을 Kubernetes 환경으로 가져오는 기능을 제공합니다. 이는 현재 다음에서 지원됩니다. `ontap-nas` , `ontap-nas-flexgroup` , `solidfire-san` , `azure-netapp-files` , 그리고 `gcp-cvs` 운전자. 이 기능은 기존 애플리케이션을 Kubernetes로 이식할 때나 재해 복구 시나리오에서 유용합니다.

ONTAP 사용할 때 `solidfire-san` 드라이버는 명령을 사용하세요 `tridentctl import volume <backend-name> <volume-name> -f /path/pvc.yaml` Trident 에서 관리할 수 있도록 기존 볼륨을 Kubernetes로 가져옵니다. import volume 명령에 사용된 PVC YAML 또는 JSON 파일은 Trident 프로비저너로 식별하는 스토리지 클래스를 가리킵니다. NetApp HCI/ SolidFire 백엔드를 사용하는 경우 볼륨 이름이 고유한지

확인하세요. 볼륨 이름이 중복된 경우 볼륨을 고유한 이름으로 복제하여 볼륨 가져오기 기능에서 구별할 수 있도록 합니다.

만약 `azure-netapp-files` 또는 `gcp-cvs` 드라이버가 사용되면 명령을 사용하세요 `tridentctl import volume <backend-name> <volume path> -f /path/pvc.yaml` 볼륨을 Kubernetes로 가져와서 Trident에서 관리하도록 합니다. 이를 통해 고유한 볼륨 참조가 보장됩니다.

위 명령을 실행하면 Trident 백엔드에서 볼륨을 찾아 크기를 읽습니다. 구성된 PVC의 볼륨 크기를 자동으로 추가하고 필요한 경우 덮어씁니다. 그런 다음 Trident 새로운 PV를 만들고 Kubernetes가 PVC를 PV에 바인딩합니다.

컨테이너가 특정 수입 PVC를 필요로 하도록 배치된 경우 PVC/PV 쌍이 볼륨 수입 프로세스를 통해 바인딩될 때까지 보류 상태로 유지됩니다. PVC/PV 쌍이 결합된 후 다른 문제가 없다면 컨테이너가 올라와야 합니다.

등록 서비스

레지스트리에 대한 저장소 배포 및 관리가 문서화되었습니다."넷앱.io" 에서"블로그" .

로깅 서비스

다른 OpenShift 서비스와 마찬가지로 로깅 서비스는 플레이북에 제공된 인벤토리 파일(호스트)에서 제공하는 구성 매개변수를 사용하여 Ansible을 사용하여 배포됩니다. 두 가지 설치 방법에 대해 설명합니다. OpenShift를 처음 설치할 때 로깅을 배포하는 방법과 OpenShift를 설치한 후에 로깅을 배포하는 방법입니다.

> Red Hat OpenShift 버전 3.9부터 공식 문서에서는 데이터 손상에 대한 우려로 인해 로깅 서비스에 NFS를 사용하지 않는 것이 좋습니다. 이는 Red Hat에서 자사 제품을 테스트한 결과를 기반으로 합니다. ONTAP NFS 서버는 이러한 문제가 없으며 로깅 배포를 쉽게 지원할 수 있습니다. 궁극적으로 로깅 서비스에 대한 프로토콜을 선택하는 것은 사용자의 몫입니다. NetApp 플랫폼을 사용할 경우 두 프로토콜 모두 잘 작동하며, NFS를 선호한다면 NFS를 피할 이유가 없습니다.

로깅 서비스와 함께 NFS를 사용하도록 선택하는 경우 Ansible 변수를 설정해야 합니다. `openshift_enable_unsupported_configurations` 에게 `true` 설치 프로그램이 실패하는 것을 방지합니다.

시작하기

로깅 서비스는 선택적으로 애플리케이션과 OpenShift 클러스터 자체의 핵심 작업에 모두 배포될 수 있습니다. 변수를 지정하여 작업 로깅을 배포하도록 선택하는 경우 `openshift_logging_use_ops` ~처럼 `true` , 서비스의 인스턴스가 두 개 생성됩니다. 작업에 대한 로깅 인스턴스를 제어하는 변수에는 "ops"가 포함되지만, 애플리케이션에 대한 인스턴스에는 포함되지 않습니다.

기본 서비스에서 올바른 스토리지를 활용하려면 배포 방법에 따라 Ansible 변수를 구성하는 것이 중요합니다. 각 배포 방법에 대한 옵션을 살펴보겠습니다.

> 아래 표에는 로깅 서비스와 관련된 저장소 구성에 관련된 변수만 포함되어 있습니다. 다른 옵션은 다음에서 찾을 수 있습니다."Red Hat OpenShift 로깅 문서" 배포에 맞게 검토, 구성 및 사용해야 합니다.

아래 표의 변수를 사용하면 Ansible 플레이북이 제공된 세부 정보를 사용하여 로깅 서비스에 대한 PV 및 PVC를 생성합니다. 이 방법은 OpenShift를 설치한 후 구성 요소 설치 플레이북을 사용하는 것보다 유연성이 상당히 떨어지지만, 기존 볼륨을 사용할 수 있는 경우에는 한 가지 옵션입니다.

변하기 쉬운	세부
`openshift_logging_storage_kind`	로 설정 `nfs` 설치 프로그램이 로깅 서비스에 대한 NFS PV를 생성하도록 합니다.
`openshift_logging_storage_host`	NFS 호스트의 호스트 이름 또는 IP 주소입니다. 이는 가상 머신의 dataLIF로 설정되어야 합니다.
`openshift_logging_storage_nfs_directory`	NFS 내보내기에 대한 마운트 경로입니다. 예를 들어, 볼륨이 다음과 같이 접합된 경우 `/openshift_logging` , 이 변수에 대해 해당 경로를 사용하게 됩니다.
`openshift_logging_storage_volume_name`	이름, 예를 들어 `pv_ose_logs` PV를 생성하려면.
`openshift_logging_storage_volume_size`	예를 들어 NFS 내보내기의 크기 `100Gi` .

OpenShift 클러스터가 이미 실행 중이고 Trident 배포 및 구성된 경우 설치 프로그램은 동적 프로비저닝을 사용하여 볼륨을 생성할 수 있습니다. 다음 변수를 구성해야 합니다.

변하기 쉬운	세부
`openshift_logging_es_pvc_dynamic`	동적으로 프로비저닝된 볼륨을 사용하려면 true로 설정합니다.
`openshift_logging_es_pvc_storage_class_name`	PVC에서 사용될 스토리지 클래스의 이름입니다.
`openshift_logging_es_pvc_size`	PVC에서 요청된 볼륨의 크기입니다.
`openshift_logging_es_pvc_prefix`	로깅 서비스에서 사용하는 PVC에 대한 접두사입니다.
`openshift_logging_es_ops_pvc_dynamic`	로 설정 `true` ops 로깅 인스턴스에 동적으로 프로비저닝된 볼륨을 사용합니다.
`openshift_logging_es_ops_pvc_storage_class_name`	ops 로깅 인스턴스의 스토리지 클래스 이름입니다.
`openshift_logging_es_ops_pvc_size`	ops 인스턴스에 대한 볼륨 요청의 크기입니다.
`openshift_logging_es_ops_pvc_prefix`	ops 인스턴스 PVC에 대한 접두사입니다.

로깅 스택 배포

초기 OpenShift 설치 프로세스의 일부로 로깅을 배포하는 경우 표준 배포 프로세스만 따르면 됩니다. Ansible은 필요한 서비스와 OpenShift 객체를 구성하고 배포하므로 Ansible이 완료되는 즉시 서비스를 사용할 수 있습니다.

하지만 초기 설치 후에 배포하는 경우 Ansible에서 구성 요소 플레이북을 사용해야 합니다. 이 프로세스는 OpenShift의 다른 버전에 따라 약간씩 변경될 수 있으므로 반드시 읽고 따르십시오."Red Hat OpenShift Container Platform 3.11 설명서" 귀하의 버전에 맞게.

메트릭 서비스

메트릭 서비스는 관리자에게 OpenShift 클러스터의 상태, 리소스 활용도, 가용성에 대한 귀중한 정보를 제공합니다. 또한 포드 자동 확장 기능에도 필요하며, 많은 조직에서 요금 청구 및/또는 쇼백 애플리케이션에 메트릭 서비스의 데이터를 사용합니다.

로깅 서비스와 OpenShift 전체와 마찬가지로 Ansible은 메트릭 서비스를 배포하는 데 사용됩니다. 또한 로깅 서비스와 마찬가지로 메트릭 서비스는 클러스터의 초기 설정 중이나 구성 요소 설치 방법을 사용하여 운영이 완료된 후에 배포할 수 있습니다. 다음 표에는 메트릭 서비스에 대한 영구 저장소를 구성할 때 중요한 변수가 포함되어 있습니다.

아래 표에는 메트릭 서비스와 관련된 스토리지 구성에 관련된 변수만 포함되어 있습니다. 배포에 맞게 검토, 구성 및 사용해야 하는 다른 옵션도 설명서에 많이 나와 있습니다.

변하기 쉬운	세부
`openshift_metrics_storage_kind`	로 설정 `nfs` 설치 프로그램이 로깅 서비스에 대한 NFS PV를 생성하도록 합니다.
`openshift_metrics_storage_host`	NFS 호스트의 호스트 이름 또는 IP 주소입니다. 이는 SVM의 dataLIF로 설정되어야 합니다.
`openshift_metrics_storage_nfs_directory`	NFS 내보내기에 대한 마운트 경로입니다. 예를 들어, 볼륨이 다음과 같이 접합된 경우 `/openshift_metrics` , 이 변수에 대해 해당 경로를 사용하게 됩니다.
`openshift_metrics_storage_volume_name`	이름, 예를 들어 `pv_ose_metrics` PV를 생성하려면.
`openshift_metrics_storage_volume_size`	예를 들어 NFS 내보내기의 크기 `100Gi` .

OpenShift 클러스터가 이미 실행 중이고 Trident 배포 및 구성된 경우 설치 프로그램은 동적 프로비저닝을 사용하여 볼륨을 생성할 수 있습니다. 다음 변수를 구성해야 합니다.

변하기 쉬운	세부
`openshift_metrics_cassandra_pvc_prefix`	PVC 지표에 사용할 접두사입니다.
`openshift_metrics_cassandra_pvc_size`	요청할 볼륨의 크기입니다.
`openshift_metrics_cassandra_storage_type`	메트릭에 사용할 저장소 유형입니다. Ansible이 적절한 저장소 클래스로 PVC를 생성하려면 이 값을 동적으로 설정해야 합니다.
`openshift_metrics_cassanda_pvc_storage_class_name`	사용할 저장 클래스의 이름입니다.

메트릭 서비스 배포

hosts/inventory 파일에 적절한 Ansible 변수를 정의한 후 Ansible을 사용하여 서비스를 배포합니다. OpenShift 설치 시점에 배포하는 경우 PV가 자동으로 생성되어 사용됩니다. 구성 요소 플레이북을 사용하여 배포하는 경우 OpenShift를 설치한 후 Ansible이 필요한 PVC를 생성하고 Trident 해당 PVC에 대한 스토리지를 프로비저닝한 후 서비스를 배포합니다.

위의 변수와 배포 프로세스는 OpenShift 버전마다 변경될 수 있습니다. 검토하고 따르세요"Red Hat의 OpenShift 배포 가이드" 귀하의 환경에 맞게 구성되도록 귀하의 버전에 맞게 구성하세요.

데이터 보호 및 재해 복구

Trident 및 Trident 사용하여 생성된 볼륨에 대한 보호 및 복구 옵션에 대해 알아보세요. 지속성 요구 사항이 있는 각 애플리케이션에 대해 데이터 보호 및 복구 전략이 있어야 합니다.

Trident 복제 및 복구

재해 발생 시 Trident 복구하기 위한 백업을 만들 수 있습니다.

Trident 복제

Trident Kubernetes CRD를 사용하여 자체 상태를 저장하고 관리하고 Kubernetes 클러스터 etcd를 사용하여 메타데이터를 저장합니다.

단계

1. Kubernetes 클러스터 etcd를 사용하여 백업합니다."Kubernetes: etcd 클러스터 백업" .
2. FlexVol volume 에 백업 아티팩트 배치

> NetApp FlexVol 있는 SVM을 다른 SVM과의 SnapMirror 관계로 보호하는 것이 좋습니다.

Trident 회수

Kubernetes CRD와 Kubernetes 클러스터 etcd 스냅샷을 사용하면 Trident 복구할 수 있습니다.

단계

1. 대상 SVM에서 Kubernetes etcd 데이터 파일과 인증서가 포함된 볼륨을 마스터 노드로 설정될 호스트에 마운트합니다.
2. Kubernetes 클러스터에 관련된 모든 필수 인증서를 아래에 복사하세요. `/etc/kubernetes/pki` 그리고 etcd 멤버 파일은 다음과 같습니다. `/var/lib/etcd` .
3. etcd 백업을 사용하여 Kubernetes 클러스터를 복원합니다."Kubernetes: etcd 클러스터 복원" .
4. 달리다 `kubectl get crd` 모든 Trident 사용자 정의 리소스가 나타났는지 확인하고 Trident 객체를 검색하여 모든 데이터를 사용할 수 있는지 확인합니다.

SVM 복제 및 복구

Trident 복제 관계를 구성할 수 없지만 스토리지 관리자는 다음을 사용할 수 있습니다. "ONTAP SnapMirror" SVM을 복제합니다.

재해가 발생하면 SnapMirror 대상 SVM을 활성화하여 데이터 제공을 시작할 수 있습니다. 시스템이 복구되면 기본으로 다시 전환할 수 있습니다.

이 작업에 관하여

SnapMirror SVM 복제 기능을 사용할 때 다음 사항을 고려하세요.

- SVM-DR이 활성화된 각 SVM에 대해 별도의 백엔드를 만들어야 합니다.
- SVM-DR을 지원하는 백엔드에 복제가 필요 없는 볼륨이 프로비저닝되는 것을 방지하기 위해 필요할 때만 복제된 백엔드를 선택하도록 스토리지 클래스를 구성합니다.
- 애플리케이션 관리자는 복제와 관련된 추가 비용과 복잡성을 이해하고 이 프로세스를 시작하기 전에 복구 계획을 신중하게 고려해야 합니다.

SVM 복제

사용할 수 있습니다"ONTAP: SnapMirror SVM 복제" SVM 복제 관계를 생성합니다.

SnapMirror 사용하면 복제할 내용을 제어하는 옵션을 설정할 수 있습니다. 수행할 때 선택한 옵션을 알아야 합니다 .Trident 사용한 SVM 복구 .

- "-동일성-참 유지"SVM 구성 전체를 복제합니다.
- "-discard-configs 네트워크"LIF 및 관련 네트워크 설정은 제외됩니다.
- "-신원-보존 거짓"볼륨과 보안 구성만 복제합니다.

Trident 사용한 SVM 복구

Trident SVM 오류를 자동으로 감지하지 않습니다. 재해가 발생하면 관리자는 수동으로 Trident 장애 조치를 새 SVM으로 시작할 수 있습니다.

단계

1. 예약된 SnapMirror 전송과 진행 중인 SnapMirror 전송을 취소하고, 복제 관계를 끊고, 소스 SVM을 중지한 다음 SnapMirror 대상 SVM을 활성화합니다.
2. 당신이 지정한 경우 `-identity-preserve false` 또는 `-discard-config network` SVM 복제를 구성할 때 다음을 업데이트하세요. `managementLIF` 그리고 `dataLIF` Trident 백엔드 정의 파일에서.
3. 확인하다 `storagePrefix` Trident 백엔드 정의 파일에 있습니다. 이 매개변수는 변경할 수 없습니다. 생략 `storagePrefix` 백엔드 업데이트가 실패하게 됩니다.
4. 다음을 사용하여 새로운 대상 SVM 이름을 반영하도록 모든 필수 백엔드를 업데이트합니다.

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

5. 당신이 지정한 경우 `-identity-preserve false` 또는 `discard-config network` 모든 애플리케이션 포드를 반송해야 합니다.

> 당신이 지정한 경우 `-identity-preserve true` Trident 에서 프로비저닝한 모든 볼륨은 대상 SVM이 활성화되면 데이터 제공을 시작합니다.

볼륨 복제 및 복구

Trident SnapMirror 복제 관계를 구성할 수 없지만 스토리지 관리자는 다음을 사용할 수 있습니다."ONTAP SnapMirror 복제 및 복구" Trident 에서 생성된 볼륨을 복제합니다.

그런 다음 복구된 볼륨을 Trident 로 가져올 수 있습니다."tridentctl 볼륨 가져오기" .

> 가져오기가 지원되지 않습니다. `ontap-nas-economy` , `ontap-san-economy` , 또는 `ontap-flexgroup-economy` 운전자.

스냅샷 데이터 보호

다음을 사용하여 데이터를 보호하고 복원할 수 있습니다.

- 영구 볼륨(PV)의 Kubernetes 볼륨 스냅샷을 생성하기 위한 외부 스냅샷 컨트롤러와 CRD입니다.

  "볼륨 스냅샷"

- ONTAP 스냅샷을 사용하면 볼륨의 전체 내용을 복원하거나 개별 파일이나 LUN을 복구할 수 있습니다.

  "ONTAP 스냅샷"

보안

보안

여기에 나열된 권장 사항을 사용하여 Trident 설치가 안전한지 확인하세요.

Trident 자체 네임스페이스에서 실행

안정적인 저장소를 보장하고 잠재적인 악성 활동을 차단하려면 애플리케이션, 애플리케이션 관리자, 사용자 및 관리 애플리케이션이 Trident 객체 정의나 포드에 액세스하지 못하도록 하는 것이 중요합니다.

다른 애플리케이션과 사용자를 Trident 에서 분리하려면 항상 Trident 자체 Kubernetes 네임스페이스에 설치하십시오 .(`trident`). Trident 자체 네임스페이스에 배치하면 Kubernetes 관리 담당자만 Trident 포드와 네임스페이스 CRD 객체에 저장된 아티팩트(해당되는 경우 백엔드 및 CHAP 비밀 등)에 액세스할 수 있습니다. 관리자만 Trident 네임스페이스에 액세스할 수 있도록 허용해야 하며 이를 통해 액세스할 수 있습니다. `tridentctl` 애플리케이션.

ONTAP SAN 백엔드에서 **CHAP** 인증 사용

Trident ONTAP SAN 워크로드에 대한 CHAP 기반 인증을 지원합니다(사용 `ontap-san` 그리고 `ontap-san-economy` 운전자). NetApp 호스트와 스토리지 백엔드 간 인증을 위해 Trident 와 함께 양방향 CHAP를 사용할 것을 권장합니다.

SAN 스토리지 드라이버를 사용하는 ONTAP 백엔드의 경우 Trident 양방향 CHAP를 설정하고 CHAP 사용자 이름과 비밀번호를 관리할 수 있습니다. `tridentctl` . 참조하다"ONTAP SAN 드라이버로 백엔드 구성을 준비합니다." Trident ONTAP 백엔드에서 CHAP를 구성하는 방법을 이해합니다.

NetApp HCI 및 **SolidFire** 백엔드에서 **CHAP** 인증 사용

NetApp 호스트와 NetApp HCI 및 SolidFire 백엔드 간의 인증을 보장하기 위해 양방향 CHAP를 배포할 것을 권장합니다. Trident 테넌트당 두 개의 CHAP 암호가 포함된 비밀 객체를 사용합니다. Trident 가 설치되면 CHAP 비밀을 관리하고 저장합니다. `tridentvolume` 해당 PV에 대한 CR 객체입니다. PV를 생성하면 Trident CHAP 비밀을 사용하여 iSCSI 세션을 시작하고 CHAP를 통해 NetApp HCI 및 SolidFire 시스템과 통신합니다.

Trident 에서 생성된 볼륨은 어떤 볼륨 액세스 그룹과도 연관되지 않습니다.

NVE 및 **NAE**와 함께 **Trident** 사용

NetApp ONTAP 디스크가 도난당하거나 반환되거나 다른 용도로 사용되는 경우 중요한 데이터를 보호하기 위해 저장

데이터 암호화를 제공합니다. 자세한 내용은 다음을 참조하세요."NetApp 볼륨 암호화 구성 개요" .

- 백엔드에서 NAE가 활성화된 경우 Trident 에서 프로비저닝된 모든 볼륨은 NAE가 활성화됩니다.
  - NVE 암호화 플래그를 설정할 수 있습니다. "" NAE 지원 볼륨을 생성합니다.
- 백엔드에서 NAE가 활성화되지 않은 경우 NVE 암호화 플래그가 설정되지 않는 한 Trident 에서 프로비저닝된 모든 볼륨은 NVE가 활성화됩니다. `false` (기본값) 백엔드 구성에서.

> NAE 지원 백엔드의 Trident 에서 생성된 볼륨은 NVE 또는 NAE로 암호화되어야 합니다.
>
> - NVE 암호화 플래그를 설정할 수 있습니다. `true` Trident 백엔드 구성에서 NAE 암호화를 재정의하고 볼륨별로 특정 암호화 키를 사용합니다.
> - NVE 암호화 플래그를 다음으로 설정 `false` NAE 지원 백엔드에서는 NAE 지원 볼륨을 생성합니다. NVE 암호화 플래그를 설정하여 NAE 암호화를 비활성화할 수 없습니다. `false` .

- NVE 암호화 플래그를 명시적으로 설정하여 Trident 에서 NVE 볼륨을 수동으로 생성할 수 있습니다. `true` .

백엔드 구성 옵션에 대한 자세한 내용은 다음을 참조하세요.

- "ONTAP SAN 구성 옵션"
- "ONTAP NAS 구성 옵션"

Linux 통합 키 설정(LUKS)

Trident 에서 ONTAP SAN 및 ONTAP SAN ECONOMY 볼륨을 암호화하기 위해 Linux Unified Key Setup(LUKS)을 활성화할 수 있습니다. Trident LUKS로 암호화된 볼륨에 대한 암호 문구 순환과 볼륨 확장을 지원합니다.

Trident 에서 LUKS 암호화 볼륨은 권장하는 대로 aes-xts-plain64 암호 및 모드를 사용합니다."미국 국립표준기술원(NIST)" .

> ASA r2 시스템에서는 LUKS 암호화가 지원되지 않습니다. ASA r2 시스템에 대한 정보는 다음을 참조하세요."ASA r2 스토리지 시스템에 대해 알아보세요" .

시작하기 전에

- 작업자 노드에는 cryptsetup 2.1 이상(3.0 미만)이 설치되어 있어야 합니다. 자세한 내용은 다음을 방문하세요 ."Gitlab: cryptsetup" .
- 성능상의 이유로 NetApp 작업자 노드가 AES-NI(Advanced Encryption Standard New Instructions)를 지원할 것을 권장합니다. AES-NI 지원을 확인하려면 다음 명령을 실행하세요.

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

아무것도 반환되지 않으면 프로세서가 AES-NI를 지원하지 않는 것입니다. AES-NI에 대한 자세한 내용은 다음을 방문하세요."Intel: 고급 암호화 표준 명령어(AES-NI)" .

LUKS 암호화 활성화

ONTAP SAN 및 ONTAP SAN ECONOMY 볼륨에 대해 Linux Unified Key Setup(LUKS)을 사용하여 볼륨별 호스트 측 암호화를 활성화할 수 있습니다.

단계

1. 백엔드 구성에서 LUKS 암호화 속성을 정의합니다. ONTAP SAN의 백엔드 구성 옵션에 대한 자세한 내용은 다음을 참조하세요."ONTAP SAN 구성 옵션" .

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

2. 사용 `parameters.selector` LUKS 암호화를 사용하여 스토리지 풀을 정의합니다. 예를 들어:

```
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. LUKS 암호문구를 포함하는 비밀번호를 생성합니다. 예를 들어:

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

제한 사항

LUKS 암호화 볼륨은 ONTAP 중복 제거 및 압축 기능을 활용할 수 없습니다.

LUKS 볼륨 가져오기를 위한 백엔드 구성

LUKS 볼륨을 가져오려면 다음을 설정해야 합니다. `luksEncryption` 에게(`true` 백엔드에서. 그만큼 `luksEncryption` 옵션은 볼륨이 LUKS 규격인지 Trident 알려줍니다.(`true`) 또는 LUKS 규격에 맞지 않음(`false`) 다음 예와 같습니다.

```
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'
```

LUKS 볼륨 가져오기를 위한 **PVC** 구성

LUKS 볼륨을 동적으로 가져오려면 주석을 설정하세요. `trident.netapp.io/luksEncryption` 에게 `true` 이 예에서 보여지는 것처럼 PVC에 LUKS 지원 스토리지 클래스를 포함합니다.

```
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
```

LUKS 암호 문구 회전

LUKS 암호를 교체하고 교체를 확인할 수 있습니다.

> 볼륨, 스냅샷 또는 비밀에서 더 이상 참조되지 않는다는 것을 확인할 때까지 암호를 잊지 마세요. 참조된 암호문구가 손실되면 볼륨을 마운트할 수 없고 데이터는 암호화되어 액세스할 수 없게 됩니다.

이 작업에 관하여

LUKS 암호 문구 순환은 새로운 LUKS 암호 문구가 지정된 후 볼륨을 마운트하는 포드가 생성될 때 발생합니다. 새로운 포드가 생성되면 Trident 볼륨의 LUKS 암호를 비밀의 활성 암호와 비교합니다.

- 볼륨의 암호가 비밀의 활성 암호와 일치하지 않으면 회전이 발생합니다.
- 볼륨의 암호가 비밀의 활성 암호와 일치하는 경우 `previous-luks-passphrase` 매개변수는 무시됩니다.

단계

1. 추가하세요 `node-publish-secret-name` 그리고 `node-publish-secret-namespace` StorageClass 매개변수. 예를 들어:

```
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. 볼륨이나 스냅샷에 있는 기존 암호를 식별합니다.

   용량

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

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

   스냅샷

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

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

3. 볼륨의 LUKS 비밀번호를 업데이트하여 새 암호와 이전 암호구를 지정합니다. 보장하다 `previous-luke-passphrase-name` 그리고 `previous-luks-passphrase` 이전 암호문구와 일치합니다.

```
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. 볼륨을 마운트하여 새로운 포드를 만듭니다. 이는 회전을 시작하는 데 필요합니다.
5. 암호가 회전되었는지 확인하세요.

   용량

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

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

스냅샷

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

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

결과

볼륨과 스냅샷에 새로운 암호문구만 반환되면 암호문구가 회전되었습니다.

> 예를 들어 두 개의 암호가 반환되는 경우 `luksPassphraseNames: ["B", "A"]` , 회전이 완료되지 않았습니다. 새로운 포드를 작동시켜 회전을 완료할 수 있습니다.

볼륨 확장 활성화

LUKS로 암호화된 볼륨에서 볼륨 확장을 활성화할 수 있습니다.

단계

1. 활성화 `CSINodeExpandSecret` 기능 게이트(베타 1.25+). 참조하다 "Kubernetes 1.25: CSI 볼륨의 노드 기반 확장을 위한 비밀 사용" 자세한 내용은.
2. 추가하세요 `node-expand-secret-name` 그리고 `node-expand-secret-namespace` StorageClass 매개변수. 예를 들어:

```
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
```

결과

온라인 스토리지 확장을 시작하면 kubelet은 드라이버에 적절한 자격 증명을 전달합니다.

Kerberos 비행 중 암호화

Kerberos 전송 중 암호화를 사용하면 관리되는 클러스터와 스토리지 백엔드 간 트래픽에 대한 암호화를 활성화하여 데이터 액세스 보안을 강화할 수 있습니다.

Trident ONTAP 스토리지 백엔드로 사용할 때 Kerberos 암호화를 지원합니다.

- 온프레미스 **ONTAP** - Trident Red Hat OpenShift 및 업스트림 Kubernetes 클러스터에서 온프레미스 ONTAP 볼륨으로의 NFSv3 및 NFSv4 연결을 통해 Kerberos 암호화를 지원합니다.

NFS 암호화를 사용하는 볼륨을 생성, 삭제, 크기 조정, 스냅샷, 복제, 읽기 전용 복제 및 가져오기할 수 있습니다.

온프레미스 **ONTAP** 볼륨을 사용하여 비행 중 **Kerberos** 암호화 구성

관리형 클러스터와 온프레미스 ONTAP 스토리지 백엔드 간의 스토리지 트래픽에 Kerberos 암호화를 활성화할 수 있습니다.

> ⓘ 온프레미스 ONTAP 스토리지 백엔드를 사용하는 NFS 트래픽에 대한 Kerberos 암호화는 다음을 통해서만 지원됩니다. `ontap-nas` 저장 드라이버.

시작하기 전에

- 귀하가 다음에 액세스할 수 있는지 확인하십시오. `tridentctl` 공익사업.
- ONTAP 스토리지 백엔드에 대한 관리자 액세스 권한이 있는지 확인하세요.
- ONTAP 스토리지 백엔드에서 공유할 볼륨의 이름을 알고 있는지 확인하세요.
- NFS 볼륨에 대한 Kerberos 암호화를 지원하도록 ONTAP 스토리지 VM을 준비했는지 확인하세요. 참조하다 "dataLIF에서 Kerberos 활성화" 지침을 보려면.
- Kerberos 암호화와 함께 사용하는 모든 NFSv4 볼륨이 올바르게 구성되었는지 확인하세요. NetApp NFSv4 도메인 구성 섹션(13페이지)을 참조하세요. "NetApp NFSv4 개선 사항 및 모범 사례 가이드" .

ONTAP 내보내기 정책 추가 또는 수정

기존 ONTAP 내보내기 정책에 규칙을 추가하거나 ONTAP 스토리지 VM 루트 볼륨과 업스트림 Kubernetes 클러스터와 공유되는 모든 ONTAP 볼륨에 대한 Kerberos 암호화를 지원하는 새로운 내보내기 정책을 만들어야 합니다. 추가하는 내보내기 정책 규칙이나 만드는 새로운 내보내기 정책은 다음과 같은 액세스 프로토콜과 액세스 권한을 지원해야 합니다.

접근 프로토콜

NFS, NFSv3, NFSv4 액세스 프로토콜을 사용하여 내보내기 정책을 구성합니다.

접근 세부 정보

볼륨에 대한 요구 사항에 따라 세 가지 Kerberos 암호화 버전 중 하나를 구성할 수 있습니다.

- **Kerberos 5** - (인증 및 암호화)
- **Kerberos 5i** - (신원 보호 기능이 있는 인증 및 암호화)
- **Kerberos 5p** - (신원 및 개인 정보 보호를 통한 인증 및 암호화)

적절한 액세스 권한으로 ONTAP 내보내기 정책 규칙을 구성합니다. 예를 들어, 클러스터가 Kerberos 5i와 Kerberos 5p 암호화를 혼합하여 NFS 볼륨을 마운트하는 경우 다음 액세스 설정을 사용합니다.

유형	읽기 전용 액세스	읽기/쓰기 액세스	슈퍼유저 접근
유닉스	활성화됨	활성화됨	활성화됨

유형	읽기 전용 액세스	읽기/쓰기 액세스	슈퍼유저 접근
케르베로스 5i	활성화됨	활성화됨	활성화됨
케르베로스 5p	활성화됨	활성화됨	활성화됨

ONTAP 내보내기 정책과 내보내기 정책 규칙을 만드는 방법에 대한 자세한 내용은 다음 문서를 참조하세요.

- "수출 정책 만들기"
- "내보내기 정책에 규칙 추가"

스토리지 백엔드 생성

Kerberos 암호화 기능을 포함하는 Trident 스토리지 백엔드 구성을 만들 수 있습니다.

이 작업에 관하여

Kerberos 암호화를 구성하는 스토리지 백엔드 구성 파일을 만들 때 다음을 사용하여 세 가지 Kerberos 암호화 버전 중 하나를 지정할 수 있습니다. `spec.nfsMountOptions` 매개변수:

- `spec.nfsMountOptions: sec=krb5`(인증 및 암호화)
- `spec.nfsMountOptions: sec=krb5i`(신원 보호 기능이 있는 인증 및 암호화)
- `spec.nfsMountOptions: sec=krb5p`(신원 및 개인 정보 보호를 통한 인증 및 암호화)

Kerberos 수준을 하나만 지정하세요. 매개변수 목록에서 두 개 이상의 Kerberos 암호화 수준을 지정하는 경우 첫 번째 옵션만 사용됩니다.

단계

1. 관리되는 클러스터에서 다음 예를 사용하여 스토리지 백엔드 구성 파일을 만듭니다. <> 괄호 안의 값을 사용자 환경의 정보로 바꾸세요.

```
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. 이전 단계에서 만든 구성 파일을 사용하여 백엔드를 만듭니다.

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

백엔드 생성에 실패하면 백엔드 구성에 문제가 있는 것입니다. 다음 명령을 실행하면 로그를 보고 원인을 파악할 수 있습니다.

```
tridentctl logs
```

구성 파일의 문제를 파악하고 수정한 후에는 create 명령을 다시 실행할 수 있습니다.

스토리지 클래스 생성

Kerberos 암호화를 사용하여 볼륨을 프로비저닝하기 위해 스토리지 클래스를 생성할 수 있습니다.

이 작업에 관하여

저장소 클래스 객체를 생성할 때 다음을 사용하여 세 가지 Kerberos 암호화 버전 중 하나를 지정할 수 있습니다. `mountOptions` 매개변수:

- `mountOptions: sec=krb5`(인증 및 암호화)
- `mountOptions: sec=krb5i`(신원 보호 기능이 있는 인증 및 암호화)
- `mountOptions: sec=krb5p`(신원 및 개인 정보 보호를 통한 인증 및 암호화)

Kerberos 수준을 하나만 지정하세요. 매개변수 목록에서 두 개 이상의 Kerberos 암호화 수준을 지정하는 경우 첫 번째 옵션만 사용됩니다. 스토리지 백엔드 구성에서 지정한 암호화 수준이 스토리지 클래스 개체에서 지정한 수준과 다른 경우 스토리지 클래스 개체가 우선합니다.

단계

1. 다음 예를 사용하여 StorageClass Kubernetes 객체를 만듭니다.

```
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. 저장 클래스를 만듭니다.

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

3. 스토리지 클래스가 생성되었는지 확인하세요.

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

다음과 비슷한 출력이 표시됩니다.

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

공급량

스토리지 백엔드와 스토리지 클래스를 만든 후 이제 볼륨을 프로비저닝할 수 있습니다. 지침은 다음을 참조하세요. "볼륨 제공" .

Azure NetApp Files 볼륨을 사용하여 진행 중인 Kerberos 암호화 구성

관리되는 클러스터와 단일 Azure NetApp Files 스토리지 백엔드 또는 Azure NetApp Files 스토리지 백엔드의 가상 풀 간의 스토리지 트래픽에 Kerberos 암호화를 활성화할 수 있습니다.

시작하기 전에

- 관리되는 Red Hat OpenShift 클러스터에서 Trident 활성화했는지 확인하세요.
- 귀하가 다음에 액세스할 수 있는지 확인하십시오. `tridentctl` 공익사업.
- 요구 사항을 확인하고 다음 지침을 따르면 Kerberos 암호화를 위한 Azure NetApp Files 저장소 백엔드가 준비되었는지 확인할 수 있습니다. "Azure NetApp Files 설명서" .
- Kerberos 암호화와 함께 사용하는 모든 NFSv4 볼륨이 올바르게 구성되었는지 확인하세요. NetApp NFSv4 도메인 구성 섹션(13페이지)을 참조하세요. "NetApp NFSv4 개선 사항 및 모범 사례 가이드" .

스토리지 백엔드 생성

Kerberos 암호화 기능을 포함하는 Azure NetApp Files 스토리지 백엔드 구성을 만들 수 있습니다.

이 작업에 관하여

Kerberos 암호화를 구성하는 스토리지 백엔드 구성 파일을 만들 때 다음 두 가지 수준 중 하나에 적용되도록 정의할 수 있습니다.

- *저장소 백엔드 수준*을 사용합니다. `spec.kerberos` 필드
- *가상 풀 레벨*을 사용하여 `spec.storage.kerberos` 필드

가상 풀 수준에서 구성을 정의하는 경우 스토리지 클래스의 레이블을 사용하여 풀이 선택됩니다.

어느 수준에서든 세 가지 Kerberos 암호화 버전 중 하나를 지정할 수 있습니다.

- `kerberos: sec=krb5`(인증 및 암호화)
- `kerberos: sec=krb5i`(신원 보호 기능이 있는 인증 및 암호화)
- `kerberos: sec=krb5p`(신원 및 개인 정보 보호를 통한 인증 및 암호화)

단계

1. 관리되는 클러스터에서 스토리지 백엔드를 정의해야 하는 위치(스토리지 백엔드 수준 또는 가상 풀 수준)에 따라 다음 예제 중 하나를 사용하여 스토리지 백엔드 구성 파일을 만듭니다. <> 괄호 안의 값을 사용자 환경의 정보로 바꾸세요.

스토리지 백엔드 수준 예제

```
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
```

가상 풀 레벨 예시

```
---
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. 이전 단계에서 만든 구성 파일을 사용하여 백엔드를 만듭니다.

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

백엔드 생성에 실패하면 백엔드 구성에 문제가 있는 것입니다. 다음 명령을 실행하면 로그를 보고 원인을 파악할 수 있습니다.

```
tridentctl logs
```

구성 파일의 문제를 파악하고 수정한 후에는 create 명령을 다시 실행할 수 있습니다.

스토리지 클래스 생성

Kerberos 암호화를 사용하여 볼륨을 프로비저닝하기 위해 스토리지 클래스를 생성할 수 있습니다.

단계

1. 다음 예를 사용하여 StorageClass Kubernetes 객체를 만듭니다.

```
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. 저장 클래스를 만듭니다.

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

3. 스토리지 클래스가 생성되었는지 확인하세요.

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

다음과 비슷한 출력이 표시됩니다.

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

공급량

스토리지 백엔드와 스토리지 클래스를 만든 후 이제 볼륨을 프로비저닝할 수 있습니다. 지침은 다음을 참조하세요. "볼륨 제공" .

저작권 정보

상표 정보