모범 사례 및 권장사항

Astra Trident

NetApp
April 16, 2024

This PDF was generated from https://docs.netapp.com/ko-kr/trident-2110/trident-reco/deploy-reco.html on April 16, 2024. Always check docs.netapp.com for the latest.

목차

모범 사례 및 권장사항 1
구축 1
스토리지 구성 1
Astra Trident 통합 9
데이터 보호 18
보안 23

모범 사례 및 권장사항

구축

Astra Trident를 배포할 때 여기에 나열된 권장 사항을 사용하십시오.

전용 네임스페이스에 구축

"네임스페이스" 서로 다른 애플리케이션 간의 관리 분리를 제공하며 리소스 공유의 장벽입니다. 예를 들어, 한 네임스페이스의 PVC는 다른 네임스페이스에서 사용할 수 없습니다. Astra Trident는 Kubernetes 클러스터의 모든 네임스페이스에 PV 리소스를 제공하고, 결과적으로 권한이 상승된 서비스 계정을 활용합니다.

또한 Trident Pod에 액세스하면 사용자가 스토리지 시스템 자격 증명 및 기타 중요한 정보에 액세스할 수 있습니다. 애플리케이션 사용자 및 관리 애플리케이션에서 Trident 객체 정의 또는 POD 자체에 액세스할 수 없도록 하는 것이 중요합니다.

할당량 및 범위 제한을 사용하여 스토리지 사용량을 제어할 수 있습니다

Kubernetes에는 2가지 기능이 있으며, 이 기능을 조합하여 애플리케이션의 리소스 사용을 제한하는 강력한 메커니즘을 제공합니다. 를 클릭합니다 "스토리지 할당량 메커니즘" 관리자가 네임스페이스별로 글로벌 및 스토리지 클래스별, 용량 및 오브젝트 수 사용 제한을 구현할 수 있도록 지원 또한 를 사용합니다 "범위 제한" 요청이 프로비저닝 사용자에게 전달되기 전에 PVC 요청이 최소값 및 최대값 내에 있는지 확인합니다.

이러한 값은 네임스페이스 단위로 정의됩니다. 즉, 각 네임스페이스에는 리소스 요구 사항에 맞는 값이 정의되어 있어야 합니다. 에 대한 자세한 내용은 여기 를 참조하십시오 "할당량을 활용하는 방법".

스토리지 구성

NetApp 포트폴리오의 각 스토리지 플랫폼은 컨테이너식으로 애플리케이션에 이점을 제공하는 고유한 기능을 제공합니다. Trident는 ONTAP, Element, E-Series의 각 주요 플랫폼에서 작동합니다. 모든 애플리케이션과 시나리오에 적합한 플랫폼이 한 개 있는 것은 아니지만, 플랫폼을 선택할 때 애플리케이션과 장치를 관리하는 팀의 요구 사항을 고려해야 합니다.

활용 중인 프로토콜을 사용하여 호스트 운영 체제의 기준 모범 사례를 따라야 합니다. 필요에 따라 특정 애플리케이션에 맞게 스토리지를 최적화할 수 있도록 백엔드, 스토리지 클래스 및 PVC 설정과 함께 사용 가능한 경우 애플리케이션 Best Practice를 통합하는 것을 고려할 수 있습니다.

ONTAP 및 **Cloud Volumes ONTAP** 모범 사례

Trident를 위한 ONTAP 및 Cloud Volumes ONTAP를 구성하기 위한 모범 사례에 대해 알아보십시오.

다음 권장 사항은 Trident에서 동적으로 프로비저닝되는 볼륨을 사용하는 컨테이너식 워크로드에 대한 ONTAP 구성 지침입니다. 각 항목을 고려하여 작업 환경의 적절성을 판단해야 합니다.

Trident 전용 **SVM**을 사용하십시오

SVM(스토리지 가상 시스템)은 ONTAP 시스템의 테넌트 간에 격리하고 관리를 제공합니다. SVM을 애플리케이션 전용으로 사용하면 권한을 위임하고 리소스 사용을 제한하는 모범 사례를 적용할 수 있습니다.

SVM 관리를 위해 몇 가지 옵션을 사용할 수 있습니다.

- 백엔드 구성에서 클러스터 관리 인터페이스를 적절한 자격 증명과 함께 제공하고 SVM 이름을 지정합니다.
- ONTAP System Manager 또는 CLI를 사용하여 SVM을 위한 전용 관리 인터페이스를 생성합니다.
- 관리 역할을 NFS 데이터 인터페이스와 공유합니다.

각 경우 인터페이스가 DNS에 있어야 하며, Trident를 구성할 때 DNS 이름을 사용해야 합니다. 이렇게 하면 네트워크 ID 보존을 사용하지 않고 SVM-DR과 같은 일부 DR 시나리오를 간편하게 수행할 수 있습니다.

SVM에 전용 또는 공유 관리 LIF를 사용하는 것이 더 이상 선호되지 않지만, 선택한 접근 방식에 맞게 네트워크 보안 정책을 조정해야 합니다. 관리 LIF는 DNS를 통해 액세스할 수 있어야 하며, 유연성을 극대화해야 합니다 "SVM-DR" Trident와 함께 사용합니다.

최대 볼륨 수를 제한합니다

ONTAP 스토리지 시스템의 최대 볼륨 수는 소프트웨어 버전과 하드웨어 플랫폼에 따라 다릅니다. 을 참조하십시오 "NetApp Hardware Universe를 참조하십시오" 정확한 제한을 결정하는 특정 플랫폼 및 ONTAP 버전. 볼륨 수가 소진되면 프로비저닝 작업이 Trident뿐 아니라 모든 스토리지 요청에 대해 실패합니다.

Trident의 'ONTAP-NAS' 및 'ONTAP-SAN' 드라이버는 생성되는 각 Kubernetes 영구 볼륨(PV)에 대해 FlexVolume을 프로비저닝합니다. 'ONTAP-NAS-이코노미' 드라이버는 200개의 PVS에 대해 약 1개의 FlexVolume을 생성합니다(50개에서 300개로 구성 가능). 'ONTAP-SAN-이코노미' 드라이버는 PVS 100대당 약 1개의 FlexVolume을 생성합니다(50개에서 200개로 구성 가능). Trident가 스토리지 시스템에서 사용 가능한 모든 볼륨을 사용하지 않도록 하려면 SVM에 제한을 설정해야 합니다. 이 작업은 명령줄에서 수행할 수 있습니다.

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

'최대 볼륨'의 값은 환경에 따라 몇 가지 기준에 따라 달라집니다.

- ONTAP 클러스터에 있는 기존 볼륨의 수입니다
- 다른 애플리케이션에 대해 Trident 외부에 프로비저닝할 것으로 예상되는 볼륨 수입니다
- Kubernetes 애플리케이션에서 사용할 것으로 예상되는 영구 볼륨의 수입니다

max-volumes 값은 개별 ONTAP 노드가 아닌 ONTAP 클러스터의 모든 노드에 프로비저닝된 총 볼륨입니다. 결과적으로, ONTAP 클러스터 노드에 다른 노드보다 훨씬 더 많은 Trident 프로비저닝 볼륨이 있을 수 있는 몇 가지 조건이 발생할 수 있습니다.

예를 들어, 2노드 ONTAP 클러스터에는 최대 2000개의 FlexVolumes를 호스팅할 수 있는 기능이 있습니다. 최대 볼륨 수를 1250으로 설정하면 매우 적절합니다. 그러나, 단 인 경우 "애그리게이트" 한 노드에서 SVM에 할당하거나, 한 노드에서 할당된 애그리게이트는 용량 등으로 인해 프로비저닝할 수 없는 경우, 다른 노드는 프로비저닝된 모든 Trident 볼륨의 타겟이 됩니다. 즉, '최대 볼륨' 값에 도달하기 전에 해당 노드에 대한 볼륨 제한에 도달하여 Trident와 해당 노드를 사용하는 다른 볼륨 작업에 영향을 줄 수 있습니다. * 클러스터의 각 노드에서 애그리게이트가 동일한 수의 Trident가 사용하는 SVM에 할당되도록 하면 이러한 상황을 방지할 수 있습니다. *

Trident에서 생성한 볼륨의 최대 크기를 제한합니다

Trident에서 생성할 수 있는 볼륨의 최대 크기를 구성하려면 'backend.json' 정의에 있는 'limitVolumeSize' 매개 변수를 사용하십시오.

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

양방향 **CHAP**를 사용하도록 **Trident**를 구성합니다

백엔드 정의에 CHAP 이니시에이터와 타겟 사용자 이름 및 암호를 지정하고 SVM에서 Trident가 CHAP를 사용하도록 설정할 수 있습니다. 백엔드 구성의 "useCHAP" 매개 변수를 사용하여 Trident는 CHAP로 ONTAP 백엔드에 대한 iSCSI 연결을 인증합니다. 양방향 CHAP 지원은 Trident 20.04 이상에서 사용할 수 있습니다.

SVM QoS 정책을 생성하고 사용합니다

SVM에 적용되는 ONTAP QoS 정책을 활용하여 Trident에서 프로비저닝된 볼륨에서 사용할 수 있는 IOPS 수를 제한합니다. 그러면 에 도움이 됩니다 "괴롭힘을 방지합니다" 또는 Trident SVM 외부의 워크로드에 영향을 주지 않는 제어 컨테이너

몇 가지 단계로 SVM에 대한 QoS 정책을 생성할 수 있습니다. 가장 정확한 정보는 사용 중인 ONTAP 버전 설명서를 참조하십시오. 아래 예는 SVM에 사용 가능한 총 IOPS를 5000으로 제한하는 QoS 정책을 생성합니다.

```
# 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 사용하는 ONTAP 버전에서 지원하는 경우에는 최소 QoS를 사용하여 컨테이너화된 워크로드에 대한 처리량을 보장하는 것을 고려할 수 있습니다. 적응형 QoS는 SVM 레벨 정책과 호환되지 않습니다.

컨테이너화된 워크로드 전용 IOPS 수는 다양한 측면에 따라 다릅니다. 그 밖의 다른 사항으로는 다음과 같은 것들이 있습니다.

- 기타 워크로드는 스토리지 어레이를 사용합니다. 스토리지 리소스를 활용하여 Kubernetes 구축과 관련되지 않은 다른 워크로드가 있는 경우, 해당 워크로드가 실수로 영향을 받지 않도록 주의해야 합니다.
- 컨테이너에서 실행 중인 예상 워크로드 IOPS 요구사항이 높은 워크로드를 컨테이너에서 실행할 경우 QoS 정책이 낮으면 잘못된 경험이 될 수 있습니다.

SVM 레벨에서 할당된 QoS 정책을 사용하면 동일한 IOPS 풀을 공유하는 SVM에 프로비저닝된 모든 볼륨이 생성된다는 점을 기억해야 합니다. 컨테이너화된 애플리케이션 중 하나 또는 그 수가 적은 경우 높은 IOPS 요구사항이 있으면 다른 컨테이너화된 워크로드에 문제가 될 수 있습니다. 이 경우 외부 자동화를 사용하여 볼륨당 QoS 정책을 할당하는 것을 고려할 수 있습니다.

ONTAP 버전이 9.8 이전인 경우 SVM * 에만 QoS 정책 그룹을 할당해야 합니다.

Trident에 대한 **QoS** 정책 그룹을 생성합니다

QoS(서비스 품질)는 경쟁 워크로드로부터 주요 워크로드의 성능이 저하되지 않도록 보장합니다. ONTAP QoS 정책 그룹은 볼륨에 대한 QoS 옵션을 제공하고 사용자가 하나 이상의 워크로드에 대한 처리량 한도를 정의할 수 있도록 지원합니다. QoS에 대한 자세한 내용은 를 참조하십시오 "QoS를 통해 처리량 보장". 백엔드에서 또는 스토리지 풀에 QoS 정책 그룹을 지정할 수 있으며, 이러한 그룹은 해당 풀 또는 백엔드에서 생성된 각 볼륨에 적용됩니다.

ONTAP에는 기존 QoS 정책과 적응형 서비스 두 가지 QoS 정책 그룹이 있습니다. 기존 정책 그룹은 IOPS 단위로 최대 또는 최소 단위의 고정 처리량을 제공합니다. 적응형 QoS는 워크로드 크기에 따라 처리량을 자동으로 확장하므로 워크로드 크기에 따라 IOPS와 TB|GB의 비율을 유지합니다. 따라서 대규모 구축 환경에서 수백 또는 수천 개의 워크로드를 관리할 경우 상당한 이점이 있습니다.

QoS 정책 그룹을 생성할 때는 다음 사항을 고려하십시오.

- 백엔드 구성의 "deefaults" 블록에 qosPolicy 키를 설정해야 합니다. 다음 백엔드 구성 예를 참조하십시오.

```
{
  "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"
      }
    }
  ]
}
```

- 각 볼륨이 정책 그룹에서 지정한 전체 처리량을 얻을 수 있도록 볼륨별로 정책 그룹을 적용해야 합니다. 공유 정책 그룹은 지원되지 않습니다.

QoS 정책 그룹에 대한 자세한 내용은 을 참조하십시오 "ONTAP 9.8 QoS 명령".

스토리지 리소스에 대한 액세스 권한을 **Kubernetes** 클러스터 구성원으로 제한합니다

Trident에서 생성한 NFS 볼륨 및 iSCSI LUN에 대한 액세스를 제한하는 것은 Kubernetes 구축을 위한 보안 환경의 중요한 구성요소입니다. 이렇게 하면 Kubernetes 클러스터의 일부가 아닌 호스트가 볼륨에 액세스하고 예기치 않게 데이터를 수정하는 것을 방지할 수 있습니다.

네임스페이스가 Kubernetes의 리소스에 대한 논리적 경계라는 것을 이해하는 것이 중요합니다. 동일한 네임스페이스의 리소스를 공유할 수 있다고 가정하지만, 특히 상호 네임스페이스 기능이 없다는 것이 중요합니다. 즉, PVS는 글로벌 객체이지만 PVC에 바인딩되면 동일한 네임스페이스에 있는 Pod에서만 액세스할 수 있습니다. * 적절한 경우 네임스페이스를 사용하여 구분을 제공하는 것이 중요합니다. *

Kubernetes 컨텍스트에서 데이터 보안과 관련하여 대부분의 조직은 컨테이너 내의 프로세스가 호스트에 마운트된 스토리지에 액세스할 수 있지만 컨테이너용 프로세스는 아닙니다. "네임스페이스" 이러한 유형의 손상을 방지하도록 설계되었습니다. 그러나 권한 있는 컨테이너에는 한 가지 예외가 있습니다.

권한 있는 컨테이너는 일반적인 것보다 훨씬 더 많은 호스트 수준 권한으로 실행되는 컨테이너입니다. 이러한 기능은 기본적으로 거부되지 않으므로 을 사용하여 기능을 사용하지 않도록 설정해야 합니다 "POD 보안 정책".

Kubernetes 및 외부 호스트 모두에서 액세스가 필요한 볼륨의 경우, Trident에서 관리하지 않고 관리자가 PV를 도입한 상태로 스토리지를 기존 방식으로 관리해야 합니다. 이렇게 하면 Kubernetes 및 외부 호스트의 연결이 모두 끊기고 볼륨을 더 이상 사용하지 않는 경우에만 스토리지 볼륨이 폐기됩니다. 또한, 맞춤형 엑스포트 정책을 적용하여 Kubernetes 클러스터 노드 및 Kubernetes 클러스터 외부의 타겟 서버에서 액세스할 수 있습니다.

전용 인프라 노드(예: OpenShift) 또는 사용자 애플리케이션을 위해 스케줄링할 수 없는 다른 노드가 있는 배포의 경우 별도의 엑스포트 정책을 사용하여 스토리지 리소스에 대한 액세스를 더욱 제한해야 합니다. 여기에는 해당 인프라 노드에 배포된 서비스(예: OpenShift Metrics 및 Logging 서비스)에 대한 엑스포트 정책과 비인프라 노드에 배포되는 표준 애플리케이션이 포함됩니다.

전용 엑스포트 정책을 사용하십시오

Kubernetes 클러스터에 있는 노드에만 액세스할 수 있도록 각 백엔드에 대한 엑스포트 정책이 있어야 합니다. Trident는 20.04 릴리스부터 내보내기 정책을 자동으로 생성하고 관리할 수 있습니다. 이러한 방법으로 Trident는 Kubernetes 클러스터의 노드에 프로비저닝되는 볼륨에 대한 액세스를 제한하고 노드 추가/삭제를 단순화합니다.

또는 수동으로 엑스포트 정책을 생성하여 각 노드 액세스 요청을 처리하는 하나 이상의 엑스포트 규칙으로 채울 수도 있습니다.

- vserver export-policy create ONTAP CLI 명령을 사용하여 엑스포트 정책을 생성합니다.
- vserver export-policy rule create ONTAP CLI 명령을 사용하여 엑스포트 정책에 규칙을 추가합니다.

이러한 명령을 실행하면 데이터에 액세스할 수 있는 Kubernetes 노드를 제한할 수 있습니다.

사용 안 함 `showmount` 애플리케이션 **SVM**을 위해

'howmount' 기능을 사용하면 NFS 클라이언트가 SVM을 쿼리하여 사용 가능한 NFS 내보내기 목록을 확인할 수 있습니다. Kubernetes 클러스터에 구축된 POD는 데이터 LIF에 대해 'howmount -e' 명령을 실행하여 액세스할 수 없는 마운트를 비롯한 사용 가능한 마운트 목록을 받을 수 있습니다. 이는 그 자체로 보안 문제가 아니라, 권한이 없는 사용자가 NFS 내보내기에 연결하는 데 도움이 될 수 있는 불필요한 정보를 제공합니다.

SVM 레벨의 ONTAP CLI 명령을 사용하여 'howmount'를 비활성화해야 합니다.

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

SolidFire 모범 사례

Trident를 위한 SolidFire 스토리지를 구성하기 위한 모범 사례에 대해 알아보십시오.

SolidFire 계정을 만듭니다

각 SolidFire 계정은 고유한 볼륨 소유자를 나타내며 자체 CHAP(Challenge-Handshake 인증 프로토콜) 자격 증명을 받습니다. 계정 이름 및 상대 CHAP 자격 증명을 사용하거나 볼륨 액세스 그룹을 통해 계정에 할당된 볼륨에 액세스할 수 있습니다. 계정에는 최대 2천 개의 볼륨이 할당될 수 있지만 볼륨은 하나의 계정에만 속할 수 있습니다.

QoS 정책을 생성합니다

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

볼륨별로 QoS 매개 변수를 설정할 수 있습니다. QoS를 정의하는 세 가지 구성 가능한 매개 변수, 즉 Min IOPS, Max IOPS, Burst IOPS를 설정하여 각 볼륨의 성능을 보장할 수 있습니다.

4KB 블록 크기에 대해 가능한 최소, 최대 및 버스트 IOPS 값입니다.

IOPS 매개 변수입니다	정의	최소 값	기본값	최대 가치**(4KB)**
최소 IOPS	볼륨에 대한 보장된 성능 수준.	50	50	15000
최대 IOPS	성능은 이 제한을 초과하지 않습니다.	50	15000	200,000
버스트 IOPS	짧은 버스트 시나리오에서 허용되는 최대 IOPS입니다.	50	15000	200,000

최대 IOPS와 버스트 IOPS는 최대 200,000으로 설정할 수 있지만, 실제 볼륨의 최대 성능은 클러스터 사용량 및 노드당 성능에 의해 제한됩니다.

블록 크기와 대역폭은 IOPS 수에 직접적인 영향을 미칩니다. 블록 크기가 증가함에 따라 시스템에서 더 큰 블록 크기를 처리하는 데 필요한 수준까지 대역폭을 높일 수 있습니다. 대역폭이 증가할수록 시스템에서 달성할 수 있는 IOPS의 수가 감소합니다. 을 참조하십시오 "SolidFire 서비스 품질" QoS 및 성능에 대한 자세한 내용은 를 참조하십시오.

SolidFire 인증

요소는 CHAP 및 vag(볼륨 액세스 그룹)의 두 가지 인증 방법을 지원합니다. CHAP는 CHAP 프로토콜을 사용하여 호스트를 백엔드에 인증합니다. 볼륨 액세스 그룹은 프로비전되는 볼륨에 대한 액세스를 제어합니다. NetApp은 CHAP를 사용하여 인증을 수행하는 것이 더 간단하고 확장 제한이 없기 때문에 CHAP를 사용하는 것이 좋습니다.

CSI 프로비저닝이 강화된 Trident는 CHAP 인증 사용을 지원합니다. VAG는 일반적인 비 CSI 작동 모드에서만 사용해야 합니다.

CHAP 인증(이니시에이터가 대상 볼륨 사용자인지 확인)은 계정 기반 액세스 제어에서만 지원됩니다. CHAP를 인증에 사용하는 경우 단방향 CHAP 및 양방향 CHAP의 두 가지 옵션을 사용할 수 있습니다. 단방향 CHAP는 SolidFire 계정 이름 및 이니시에이터 암호를 사용하여 볼륨 액세스를 인증합니다. 양방향 CHAP 옵션은 볼륨이 계정 이름과 이니시에이터 암호를 통해 호스트를 인증한 다음 호스트가 계정 이름과 타겟 암호를 통해 볼륨을 인증하기 때문에

볼륨을 인증하는 가장 안전한 방법을 제공합니다.

그러나 CHAP를 설정할 수 없고 VAG가 필요한 경우 액세스 그룹을 생성하고 호스트 이니시에이터 및 볼륨을 액세스 그룹에 추가합니다. 액세스 그룹에 추가하는 각 IQN은 CHAP 인증을 사용하거나 사용하지 않고 그룹의 각 볼륨에 액세스할 수 있습니다. iSCSI 이니시에이터가 CHAP 인증을 사용하도록 구성된 경우 계정 기반 액세스 제어가 사용됩니다. iSCSI 초기자가 CHAP 인증을 사용하도록 구성되지 않은 경우 볼륨 액세스 그룹 액세스 제어가 사용됩니다.

E-Series 모범 사례

Trident를 위한 E-Series 스토리지를 구성하기 위한 모범 사례에 대해 알아보십시오.

E-Series 디스크 풀 및 볼륨 그룹

요구 사항에 따라 디스크 풀 및 볼륨 그룹을 생성하고 전체 스토리지 용량을 볼륨으로 구성하고 호스트 간에 공유하는 방법을 결정합니다. 디스크 풀과 볼륨 그룹은 모두 논리적으로 그룹화되고 애플리케이션 호스트에 하나 이상의 볼륨을 제공하는 드라이브 세트로 구성되어 있습니다. 디스크 풀 또는 볼륨 그룹의 모든 드라이브는 동일한 미디어 유형이어야 합니다.

E-Series 호스트 그룹

Trident는 호스트 그룹을 사용하여 프로비저닝하는 볼륨(LUN)에 액세스합니다. 기본적으로 Trident는 구성에 다른 호스트 그룹 이름을 지정하지 않는 한 "트리덴트"라는 호스트 그룹을 사용합니다. Trident는 자체적으로 호스트 그룹을 생성하거나 관리하지 않습니다. Trident에서 E-Series 스토리지 백엔드를 설정하기 전에 호스트 그룹을 생성해야 합니다. 모든 Kubernetes 작업자 노드 iSCSI IQN 이름이 호스트 그룹에서 업데이트되었는지 확인합니다.

E-Series의 스냅샷 스케줄입니다

필요한 간격으로 볼륨 백업을 수행할 수 있도록 스냅샷 일정을 생성하고 Trident에서 생성한 볼륨을 스냅샷 일정에 할당합니다. 스냅샷 정책에 따라 수행된 스냅샷을 기반으로 스냅샷 이미지를 기본 볼륨으로 복원하여 볼륨에 대해 롤백 작업을 수행할 수 있습니다. SANtricity 시스템 관리자를 사용하여 스냅샷 스케줄을 생성해야 합니다.

스냅샷 정합성 보장 그룹

스냅샷 정합성 보장 그룹을 설정하는 것도 여러 볼륨에 걸쳐 있는 애플리케이션에 적합합니다. 정합성 보장 그룹의 목적은 여러 볼륨의 동시 스냅샷 이미지를 만들어 특정 시점에 볼륨 컬렉션 복제본의 정합성이 보장되도록 하는 것입니다. SANtricity 시스템 관리자를 사용하여 일관성 그룹을 생성해야 합니다.

Cloud Volumes Service for AWS 모범 사례

Trident를 위한 AWS에서 Cloud Volumes Service를 구성하기 위한 모범 사례에 대해 알아보십시오.

엑스포트 정책을 생성합니다

승인된 노드 집합만 Cloud Volumes Service를 통해 프로비저닝된 볼륨에 액세스할 수 있도록 하려면 Cloud Volumes Service를 생성하는 동안 엑스포트 정책에 대한 적절한 규칙을 설정합니다. Trident를 통해 Cloud Volume Services에서 볼륨을 프로비저닝할 때 백엔드 파일의 'exportRule' 매개 변수를 사용하여 필요한 Kubernetes 노드에 대한 액세스를 제공해야 합니다.

스냅샷 정책을 생성합니다

Cloud Volume Service를 통해 프로비저닝된 볼륨에 대한 스냅샷 정책을 생성하여 스냅샷이 필요한 간격으로 생성되도록 합니다. 이렇게 하면 일정한 간격으로 데이터를 백업할 수 있으며 데이터 손실이나 데이터 손상 시 데이터를 복원할 수 있습니다. 볼륨 세부 정보 페이지에서 적절한 스케줄을 선택하여 Cloud Volume Service에서 호스팅되는 볼륨에 대한 스냅샷 정책을 설정할 수 있습니다.

적절한 서비스 수준, 스토리지 용량 및 스토리지 대역폭을 선택합니다

Cloud Volume Services for AWS는 표준, 프리미엄, 익스트림 등 다양한 서비스 수준을 제공합니다. 이러한 서비스 수준은 다양한 스토리지 용량 및 스토리지 대역폭 요구 사항을 충족합니다. 비즈니스 요구 사항에 따라 적절한 서비스 수준을 선택해야 합니다.

애플리케이션의 특정 요구 사항에 따라 볼륨을 생성하는 동안 할당된 스토리지의 필요한 크기를 선택해야 합니다. 할당된 스토리지를 결정할 때는 다음 두 가지 요소를 고려해야 합니다.

- 특정 애플리케이션의 스토리지 요구 사항
- 피크 또는 에지에 필요한 대역폭입니다

스토리지 대역폭은 선택한 서비스 수준과 할당된 용량의 조합에 따라 달라집니다. 따라서 필요한 대역폭을 염두에 두고 적절한 서비스 수준과 할당된 용량을 선택합니다.

Trident에서 생성한 볼륨의 최대 크기를 제한합니다

백엔드 구성 파일의 'limitVolumeSize' 매개 변수를 사용하여 AWS용 Cloud Volume Services의 Trident에서 생성한 볼륨의 최대 크기를 제한할 수 있습니다. 이 매개 변수를 설정하면 요청된 볼륨 크기가 설정된 값을 초과할 경우 프로비저닝이 실패합니다.

자세한 정보는 어디서 찾을 수 있습니까?

다음은 몇 가지 모범 사례 문서입니다. 를 검색합니다 "NetApp 라이브러리" 최신 버전의 경우.

- ONTAP *
- "NFS Best Practice and Implementation Guide를 참조하십시오"
- "SAN 관리 가이드를 참조하십시오" (iSCSI의 경우)
- "RHEL용 iSCSI Express 구성"

Element 소프트웨어 *

- "Linux용 SolidFire 구성"
- NetApp HCI *
- "NetApp HCI 구축 사전 요구 사항"
- "NetApp 배포 엔진에 액세스합니다"

E-Series * 를 통해 비즈니스 운영

- "Linux용 설치 및 구성"
- 응용 프로그램 모범 사례 정보 *

- "ONTAP 기반 MySQL의 모범 사례"
- "SolidFire 기반 MySQL의 모범 사례"
- "NetApp SolidFire 및 Cassandra"
- "SolidFire에 대한 Oracle 모범 사례"
- "SolidFire에 대한 PostgreSQL Best Practice"

모든 애플리케이션에 구체적인 지침이 있는 것은 아니며 NetApp 팀과 함께 을 사용하는 것이 중요합니다 "NetApp 라이브러리" 최신 설명서를 참조하십시오.

Astra Trident 통합

Astra Trident를 통합하려면 다음과 같은 설계 및 아키텍처 요소를 통합해야 합니다. 드라이버 선택 및 배포, 스토리지 클래스 설계, 가상 스토리지 풀 설계, PVC(Persistent Volume Claim)는 Astra Trident를 사용한 스토리지 프로비저닝, 볼륨 운영, OpenShift 서비스 구축에 영향을 줍니다.

운전자 선택 및 전개

ONTAP용 백엔드 드라이버를 선택합니다

ONTAP 시스템에는 4개의 서로 다른 백엔드 드라이버를 사용할 수 있습니다. 이러한 드라이버는 사용 중인 프로토콜과 스토리지 시스템에서 볼륨을 프로비저닝하는 방법에 따라 다릅니다. 따라서 어떤 드라이버를 배포할지 신중하게 고려해야 합니다.

상위 레벨에서는 애플리케이션에 공유 스토리지가 필요한 구성 요소(동일한 PVC에 액세스하는 여러 Pod)가 있는 경우 NAS 기반 드라이버가 기본 선택이고 블록 기반 iSCSI 드라이버는 비공유 스토리지의 요구를 충족합니다. 애플리케이션의 요구사항 및 스토리지 및 인프라 팀의 편안함 수준을 기준으로 프로토콜을 선택합니다. 일반적으로 대부분의 애플리케이션에서 두 서버 간에 차이가 거의 없기 때문에 공유 스토리지(둘 이상의 POD에 동시 액세스가 필요한 경우)가 필요한지 여부에 따라 결정하는 경우가 많습니다.

ONTAP 백 엔드에 대한 5개의 드라이버가 아래에 나열되어 있습니다.

- ONTAP-NAS: 프로비저닝되는 각 PV는 ONTAP FlexVolume입니다.
- ONTAP-NAS-이코노미: 각 PV 프로비저닝은 qtree이며 FlexVolume당 qtree(기본값 200)를 구성할 수 있습니다.
- ONTAP-NAS-flexgroup: 각 PV는 전체 ONTAP FlexGroup로 프로비저닝되고 SVM에 할당된 모든 애그리게이트가 사용됩니다.
- ONTAP-SAN: 각 PV는 자체 FlexVolume 내의 LUN입니다.
- ONTAP-SAN-이코노미: 각 PV는 LUN으로 프로비저닝되며 FlexVolume당 구성 가능한 LUN 수(기본값 100)가 있습니다.

세 개의 NAS 드라이버 중 하나를 선택할 경우 해당 기능에 약간의 영향을 줍니다. 이 기능은 응용 프로그램에서 사용할 수 있습니다.

아래 표에서 모든 기능이 Astra Trident를 통해 표시되는 것은 아닙니다. 용량 할당 후 기능을 적용하려면 스토리지 관리자가 일부 기능을 적용해야 합니다. 위 첨자 각주는 기능 및 드라이버별 기능을 구별합니다.

ONTAP NAS 드라이버	스냅샷 수	복제	동적 엑스포트 정책	다중 연결	**QoS**를 참조하십시오	크기 조정	복제
'ONTAP-NAS'	예	예	Yesfootnote: 5[]	예	Yesfootnote: 1[]	예	Yesfootnote: 1[]
ONTAP-NAS-이코노미	Yesfootnote: 3[]	Yesfootnote: 3[]	Yesfootnote: 5[]	예	Yesfootnote: 3[]	예	Yesfootnote: 3[]
'ONTAP-NAS-Flexgroup'	Yesfootnote: 1[]	아니요	Yesfootnote: 5[]	예	Yesfootnote: 1[]	예	Yesfootnote: 1[]

Astra Trident는 ONTAP용 SAN 드라이버 2개를 제공하며 해당 기능은 아래에 나와 있습니다.

ONTAP SAN 드라이버	스냅샷 수	복제	다중 연결	양방향 **CHAP**	**QoS**를 참조하십시오	크기 조정	복제
'ONTAP-SAN'	예	예	Yesfootnote: 4[]	예	Yesfootnote: 1[]	예	Yesfootnote: 1[]
ONTAP-SAN-이코노미	예	예	Yesfootnote: 4[]	예	Yesfootnote: 3[]	Yesfootnote: 1[]	Yesfootnote: 3[]

> 위 표의 각주: Yes [1]: Astra Trident에서 관리하지 않음 Yes [2]: Astra Trident에서 관리하지만 PV 세분화되지는 않음 Yes [3]: Astra Trident에서 관리하지 않음, PV 세분화됨 Yes [4]: 원시 블록 볼륨에서 지원됨 Yes [5]: CSI Trident에서 지원

PV 세분화되지 않은 기능은 전체 FlexVolume에 적용되고 모든 PVS(즉, 공유 FlexVol의 qtree 또는 LUN)는 공통 스케줄을 공유합니다.

위 표에서 볼 수 있듯이, ONTAP-NAS와 ONTAP-NAS-이코노미 간의 기능 대부분은 동일합니다. 그러나 ONTAP-NAS-이코노미 드라이버는 PV 단위의 일정 제어 기능을 제한하므로 특히 재해 복구 및 백업 계획에 영향을 줄 수 있습니다. ONTAP 스토리지에서 PVC 클론 기능을 활용하고자 하는 개발팀은 ONTAP-NAS, ONTAP-SAN 또는 ONTAP-SAN 절약 드라이버를 사용할 때만 가능합니다.

> 좋idfire-san의 드라이버도 PVC를 클로닝할 수 있습니다.

Cloud Volumes ONTAP용 백엔드 드라이버를 선택합니다

Cloud Volumes ONTAP은 NAS 및 SAN 프로토콜(NFS, SMB/CIFS 및 iSCSI)을 지원하는 파일 공유 및 블록 레벨 스토리지 등 다양한 활용 사례에 맞게 엔터프라이즈급 스토리지 기능과 함께 데이터 제어를 제공합니다. Cloud Volume ONTAP와 호환되는 드라이버는 ONTAP-NAS, ONTAP-NAS-이코노미, ONTAP-SAN, ONTAP-SAN 경제입니다. Cloud Volume ONTAP for AWS, Cloud Volume ONTAP for Azure, Cloud Volume ONTAP for GCP에 적용할 수 있습니다.

ONTAP용 **Amazon FSx**의 백엔드 드라이버를 선택합니다

ONTAP용 Amazon FSx를 사용하면 고객이 익숙한 NetApp 기능, 성능 및 관리 기능을 활용하는 동시에, AWS에 데이터를 저장하는 간편성, 민첩성, 보안, 확장성을 활용할 수 있습니다. ONTAP용 FSX는 ONTAP의 다양한 파일 시스템 기능과 관리 API를 지원합니다. Cloud Volume ONTAP와 호환되는 드라이버는 ONTAP-NAS, ONTAP-NAS-이코노미, ONTAP-NAS-Flexgroup, ONTAP-SAN, ONTAP-SAN입니다.

NetApp HCI/SolidFire의 백엔드 드라이버를 선택합니다

NetApp HCI/SolidFire 플랫폼과 함께 사용되는 'olidfire-SAN' 드라이버는 관리자가 QoS 제한을 기반으로 Trident에 대한 Element 백엔드를 구성하는 데 도움이 됩니다. Trident에서 프로비저닝한 볼륨에 대한 특정 QoS 제한을 설정하기 위해 백엔드를 설계하려면 백엔드 파일에 'type' 매개 변수를 사용하십시오. 또한 관리자는 'limitVolumeSize' 매개 변수를 사용하여 스토리지에 생성할 수 있는 볼륨 크기를 제한할 수 있습니다. 현재 볼륨 크기 조정 및 볼륨 복제와 같은 Element 스토리지 기능은 'olidfire-SAN' 드라이버를 통해 지원되지 않습니다. 이러한 작업은 Element 소프트웨어 웹 UI를 통해 수동으로 수행해야 합니다.

SolidFire 드라이버	스냅샷 수	복제	다중 연결	**CHAP**	**QoS**를 참조하십시오	크기 조정	복제
'솔더불-산'	예	예	Yes [2]	예	예	예	Yesfootnote: 1[]

> 각주: Yesfootnote: 1 [ ]: Astra Trident Yesfootnote: 2 [ ]: 원시 블록 볼륨에서 지원됩니다

Azure NetApp Files용 백엔드 드라이버를 선택합니다

Astra Trident는 'Azure-NetApp-files' 드라이버를 사용하여 를 관리합니다 "Azure NetApp Files" 서비스.

이 드라이버 및 구성 방법에 대한 자세한 내용은 에서 찾을 수 있습니다 "Azure NetApp Files를 위한 Astra Trident 백엔드 구성입니다".

Azure NetApp Files 드라이버	스냅샷 수	복제	다중 연결	**QoS**를 참조하십시오	비즈니스	복제
'Azure-NetApp-파일'	예	예	예	예	예	Yesfootnote: 1[]

> 각주: Yesfootnote: 1 [ ]: Astra Trident에서 관리하지 않습니다

Cloud Volumes Service with AWS의 백엔드 드라이버를 선택합니다

Astra Trident는 AWS-CVS 드라이버를 사용하여 AWS 백엔드의 Cloud Volumes Service와 연결합니다. Trident에서 AWS 백엔드를 구성하려면 백엔드 파일에 apiRegion, apiURL, apiKey, s ecretKey를 지정해야 합니다. 이러한 값은 CVS 웹 포털의 계정 설정/API 액세스에서 확인할 수 있습니다. 지원되는 서비스 수준은 CVS에 맞춰 조정되며 '표준', '프리미엄', '익스트림'도 포함됩니다. 현재 100G는 프로비저닝할 최소 볼륨 크기입니다. CVS의 향후 릴리스에서는 이 제한이 적용되지 않을 수 있습니다.

AWS 드라이버에 대한 **CV**	스냅샷 수	복제	다중 연결	**QoS**를 참조하십시오	비즈니스	복제
'AWS-CV'	예	예	예	예	예	Yesfootnote: 1[]

> 각주: Yesfootnote: 1 [ ]: Astra Trident에서 관리하지 않습니다

AWS-CV 드라이버는 가상 스토리지 풀을 사용합니다. 가상 스토리지 풀은 백엔드를 추상화하여 Trident가 볼륨 배치를 결정할 수 있도록 합니다. 관리자는 backend.json 파일에 있는 가상 스토리지 풀을 정의합니다. 스토리지 클래스는 레이블을 사용하여 가상 스토리지 풀을 식별합니다.

Cloud Volumes Service with GCP의 백엔드 드라이버를 선택합니다

Astra Trident는 GCP-cvs 드라이버를 사용하여 GCP 백엔드의 Cloud Volumes Service와 연결합니다. Trident에서 GCP 백엔드를 구성하려면 백엔드 파일에 projectNumber, apiRegion 및 apiKey를 지정해야 합니다. GCP 웹 포털에서 프로젝트 번호를 확인할 수 있으며, GCP에서 Cloud Volumes에 대한 API 액세스를 설정하는 동안 생성한 서비스 계정 프라이빗 키 파일에서 API 키를 가져와야 합니다. Astra Trident는 두 가지 중 하나로 CVS 볼륨을 생성할 수 있습니다 "서비스 유형":

1. * CVS *: 기본 CVS 서비스 유형으로, 제한된/중간 수준의 성능으로 높은 조널 가용성을 제공합니다.
2. * CVS - 성능 *: 성능이 중요한 운영 워크로드에 가장 적합한 성능 최적화 서비스 유형입니다. 세 가지 서비스 수준('Standard', 'Premium', 'Extreme')을 선택할 수 있습니다. 현재 100GiB는 프로비저닝할 최소 CVS 성능 볼륨 크기이고 CVS 볼륨은 300GiB 이상이어야 합니다. CVS의 향후 릴리스에서는 이 제한이 적용되지 않을 수 있습니다.

> 기본 CVS 서비스 유형 ['storageClass=software']를 사용하여 백엔드를 배포할 때 사용자 * 는 해당 프로젝트 번호 및 프로젝트 ID에 대해 GCP의 1TiB 미만의 볼륨 기능에 대한 액세스 * 를 얻어야 합니다. Trident에서 1TiB 미만의 볼륨을 프로비저닝하는 데 이 작업이 필요합니다. 그렇지 않은 경우, 600GiB 미만의 PVC에 대해 체적 생성 * 이 실패합니다. 사용 "이 양식입니다" 1TiB 미만의 볼륨에 대한 액세스 권한 얻기

GCP 드라이버에 대한 **CVS**	스냅샷 수	복제	다중 연결	**QoS**를 참조하십시오	비즈니스	복제
GCP-CV	예	예	예	예	예	Yesfootnote: 1[]

> 각주: Yesfootnote: 1 [ ]: Astra Trident에서 관리하지 않습니다

GCP-CV 드라이버는 가상 스토리지 풀을 사용합니다. 가상 스토리지 풀은 백엔드를 추상화하여 Astra Trident가 볼륨 배치를 결정할 수 있도록 합니다. 관리자는 backend.json 파일에 있는 가상 스토리지 풀을 정의합니다. 스토리지 클래스는 레이블을 사용하여 가상 스토리지 풀을 식별합니다.

스토리지 클래스 설계

Kubernetes Storage Class 객체를 생성하려면 개별 스토리지 클래스를 구성 및 적용해야 합니다. 이 섹션에서는 애플리케이션에 대한 스토리지 클래스를 설계하는 방법에 대해 설명합니다.

특정 백엔드 사용률을 위한 스토리지 클래스 설계

특정 스토리지 클래스 객체 내에서 필터링을 사용하여 해당 스토리지 클래스에 사용할 스토리지 풀 또는 풀 세트를 결정할 수 있습니다. Storage Class에서 'toragePools', 'additionalStoragePools', 'excludeStoragePools' 등의 세 가지 필터를 설정할 수 있습니다.

'toragePools' 매개 변수는 지정된 속성과 일치하는 풀 세트로 스토리지를 제한하는 데 도움이 됩니다. 추가 StoragePools 매개변수는 Astra Trident가 프로비저닝에 사용할 풀 세트를 속성 및 'toragePools' 매개 변수로 선택한 풀 세트와 함께 확장하는 데 사용됩니다. 매개 변수만 사용하거나 둘 모두를 함께 사용하여 적절한 스토리지 풀 세트가 선택되었는지 확인할 수 있습니다.

excludeStoragePools 매개 변수는 속성과 일치하는 나열된 풀 세트를 명시적으로 제외하는 데 사용됩니다.

QoS 정책을 에뮬레이트하기 위한 스토리지 클래스 설계

서비스 품질 정책을 에뮬레이트하기 위해 스토리지 클래스를 설계하려면 '미디어' 속성을 HDD 또는 'SSD'로 사용하여 스토리지 클래스를 생성합니다. 스토리지 클래스에 언급된 미디어 특성에 따라, Trident는 미디어 속성과 일치하도록 HDD 또는 SSD 애그리게이트를 제공하는 적절한 백엔드를 선택한 다음 볼륨 프로비저닝을 특정 애그리게이트로 전달합니다. 따라서 프리미엄 QoS 정책으로 분류될 수 있는 '미디어' 속성이 'SD'로 설정된 스토리지 클래스 Premium을 생성할 수 있습니다. 표준 QoS 정책으로 분류될 수 있는 미디어 속성을 'HDD'로 설정하는 또 다른 스토리지 클래스 표준을 생성할 수 있습니다. 또한 스토리지 클래스에서 ""IOPS"" 속성을 사용하여 QoS 정책으로 정의할 수 있는 Element 어플라이언스로 프로비저닝을 리디렉션할 수도 있습니다.

특정 기능을 기반으로 백엔드를 활용하는 스토리지 클래스 설계

스토리지 클래스는 씬 및 일반 프로비저닝, 스냅샷, 클론 및 암호화와 같은 기능이 설정된 특정 백엔드에서 볼륨 프로비저닝을 수행하도록 설계되었습니다. 사용할 스토리지를 지정하려면 필요한 기능이 설정된 적절한 백엔드를 지정하는 스토리지 클래스를 생성합니다.

가상 스토리지 풀을 위한 스토리지급 설계

모든 Astra Trident 백엔드에 가상 스토리지 풀을 사용할 수 있습니다. Astra Trident가 제공하는 드라이버를 사용하여 백엔드에 대한 가상 스토리지 풀을 정의할 수 있습니다.

가상 스토리지 풀을 사용하면 관리자가 저장소 클래스를 통해 참조할 수 있는 백엔드에 대한 추상화 수준을 생성할 수 있으므로 백엔드에 볼륨을 보다 유연하고 효율적으로 배치할 수 있습니다. 동일한 서비스 클래스로 다른 백엔드를 정의할 수 있습니다. 또한 동일한 백엔드에서 여러 스토리지 풀을 생성할 수 있지만 특성이 다릅니다. 특정 레이블이 있는 선택기로 스토리지 클래스를 구성한 경우 Astra Trident는 볼륨을 배치할 모든 선택기 레이블과 일치하는 백엔드를 선택합니다. 스토리지 클래스 선택기 레이블이 여러 스토리지 풀과 일치하면 Astra Trident가 볼륨 용량을 할당할 스토리지 풀 중 하나를 선택합니다.

가상 스토리지 풀 설계

백엔드를 생성하는 동안 일반적으로 매개 변수 집합을 지정할 수 있습니다. 관리자가 동일한 스토리지 자격 증명을 사용하여 다른 매개 변수 집합을 가진 다른 백엔드를 생성할 수 없었습니다. 가상 스토리지 풀의 도입으로 이 문제가 완화되었습니다. 가상 스토리지 풀은 백엔드 및 Kubernetes 스토리지 클래스 간에 도입된 레벨 추상화입니다. 따라서 관리자는 Kubernetes 스토리지 클래스를 통해 백엔드에 독립적인 방식으로 Selector로 참조할 수 있는 레이블과 함께 매개 변수를 정의할 수 있습니다. Astra Trident를 사용하여 지원되는 모든 NetApp 백엔드에 대해 가상 스토리지 풀을 정의할 수 있습니다. 해당 목록에는 SolidFire/NetApp HCI, ONTAP, Cloud Volumes Service on AWS, GCP 및 Azure NetApp Files가 포함됩니다.

> 가상 스토리지 풀을 정의할 때는 백엔드 정의에서 기존 가상 풀의 순서를 재정렬하지 않는 것이 좋습니다. 또한 기존 가상 풀의 속성을 편집/수정하고 대신 새 가상 풀을 정의하는 것이 좋습니다.

다양한 서비스 수준/**QoS**를 에뮬레이트할 수 있도록 가상 스토리지 풀을 설계합니다

서비스 클래스를 에뮬레이트하기 위해 가상 스토리지 풀을 설계할 수 있습니다. Cloud Volume Service for AWS를 위한 가상 풀 구축을 사용하여 다양한 서비스 클래스를 설정하는 방법을 살펴봅시다. 다양한 성능 수준을 나타내는 여러 레이블을 사용하여 AWS-CVS 백엔드를 구성합니다. '레벨 11'을 적절한 성과 수준으로 설정하고 각 레이블 아래에 다른 필수 요소를 추가합니다. 이제 다른 가상 스토리지 풀에 매핑할 다른 Kubernetes 스토리지 클래스를 생성합니다. parameters.selector` 필드를 사용하여 각 StorageClass는 볼륨을 호스팅하는 데 사용할 수 있는 가상 풀을 호출합니다.

특정 측면을 할당할 수 있도록 가상 풀을 설계합니다

특정 측면의 여러 가상 스토리지 풀을 단일 스토리지 백엔드에서 설계할 수 있습니다. 이를 위해 백엔드에 여러 레이블을 구성하고 각 레이블 아래에 필요한 측면을 설정합니다. 이제 다른 가상 스토리지 풀에 매핑될 ' parameters.selector` ' 필드를 사용하여 다른 Kubernetes 스토리지 클래스를 만들 수 있습니다. 백엔드에서 프로비저닝되는 볼륨에는 선택한 가상 스토리지 풀에 정의된 측면이 있습니다.

스토리지 프로비저닝에 영향을 미치는 **PVC** 특성

요청된 스토리지 클래스 이외의 일부 매개 변수는 PVC를 생성할 때 Astra Trident의 프로비저닝 결정 프로세스에 영향을 줄 수 있습니다.

액세스 모드

PVC를 통한 저장 요청 시 필수 필드 중 하나가 액세스 모드입니다. 원하는 모드는 스토리지 요청을 호스팅하기 위해 선택한 백엔드에 영향을 줄 수 있습니다.

Astra Trident는 다음 매트릭스에 따라 지정된 액세스 방법과 사용된 스토리지 프로토콜을 일치시키려고 시도합니다. 이는 기본 스토리지 플랫폼과 무관합니다.

	ReadWriteOnce 를 참조하십시오	**ReadOnlyMany** 를 참조하십시오	**ReadWriteMany** 를 참조하십시오
iSCSI	예	예	예(원시 블록)
NFS 를 참조하십시오	예	예	예

NFS 백엔드가 구성되지 않은 상태로 Trident 배포에 제출된 ReadWriteMany PVC에 대한 요청은 볼륨이 프로비저닝되지 않습니다. 이러한 이유로 요청자는 자신의 응용 프로그램에 적합한 액세스 모드를 사용해야 합니다.

볼륨 작업입니다

영구 볼륨 수정

영구 볼륨은 Kubernetes에서 두 가지 예외, 영구적 객체입니다. 생성된 후에는 부가세 반환 청구액 정책 및 크기를 수정할 수 있습니다. 그러나 이렇게 해서 Kubernetes 외부에서 볼륨의 일부 측면이 수정되지 않도록 할 수는 없습니다. 특정 애플리케이션에 맞게 볼륨을 사용자 지정하거나, 실수로 용량이 소비되지 않도록 하거나, 어떠한 이유로든 볼륨을 다른 스토리지 컨트롤러로 이동하는 것이 좋을 수 있습니다.

> 현재 Kubernetes 트리 프로비저닝 시 NFS 또는 iSCSI PVS의 볼륨 크기 조정 작업은 지원되지 않습니다. Astra Trident는 NFS 및 iSCSI 볼륨 확장을 지원합니다.

PV의 접속 세부 정보는 생성 후 수정할 수 없습니다.

주문형 볼륨 스냅샷을 생성합니다

Astra Trident는 CSI 프레임워크를 사용하여 필요 시 볼륨 스냅샷 생성 및 스냅샷에서 PVC 생성을 지원합니다. 스냅샷은 편리한 데이터 시점 복사본을 유지 관리하는 방법을 제공하며 Kubernetes의 소스 PV와 독립적인 라이프사이클을 갖고 있습니다. 이러한 스냅샷을 사용하여 PVC를 복제할 수 있습니다.

스냅샷으로부터 볼륨을 생성합니다

Astra Trident는 볼륨 스냅샷으로부터 PersistentVolumes 생성을 지원합니다. 이를 위해 PersistentVolumeClaim을 생성하고 볼륨을 생성해야 하는 필수 스냅샷으로 "소스"를 언급하기만 하면 됩니다. Astra Trident는 스냅샷에 데이터가 있는 볼륨을 생성하여 이 PVC를 처리합니다. 이 기능을 사용하면 지역 간에 데이터를 복제하거나 테스트 환경을 생성하거나 손상되거나 손상된 운영 볼륨을 전체적으로 교체하거나 특정 파일 및 디렉토리를 검색하여 연결된 다른 볼륨으로 전송할 수 있습니다.

클러스터에서 볼륨 이동

스토리지 관리자는 ONTAP 클러스터의 Aggregate와 컨트롤러 간에 볼륨을 스토리지 소비자로 중단 없이 이동할 수 있습니다. 대상 애그리게이트는 Astra Trident가 사용하는 SVM이 액세스할 수 있는 경우, 이 작업은 Astra Trident 또는 Kubernetes 클러스터에 영향을 주지 않습니다. 여기서 중요한 점은 애그리게이트를 SVM에 새로 추가한 경우, Astra Trident에 다시 추가하여 백엔드를 새로 고쳐야 한다는 것입니다. 그러면 Astra Trident가 SVM의 인벤토리를 다시 만들어 새 애그리게이트를 인식할 수 있습니다.

그러나 Astra Trident는 백엔드에서 볼륨을 이동하는 기능을 자동으로 지원하지 않습니다. 여기에는 동일한 클러스터, 클러스터 간 또는 다른 스토리지 플랫폼(스토리지 시스템이 Astra Trident에 연결된 SVM인 경우에도 해당 스토리지 플랫폼)에 있는 SVM이 포함됩니다.

볼륨이 다른 위치에 복사되면 볼륨 가져오기 기능을 사용하여 현재 볼륨을 Astra Trident로 가져올 수 있습니다.

볼륨 확장

Astra Trident는 NFS 및 iSCSI PVS 크기를 조정할 수 있도록 지원합니다. 따라서 사용자는 Kubernetes 계층을 통해 직접 볼륨의 크기를 조정할 수 있습니다. ONTAP, SolidFire/NetApp HCI 및 Cloud Volumes Service 백엔드를 포함한 모든 주요 NetApp 스토리지 플랫폼에서 볼륨 확장이 가능합니다. 나중에 확장을 허용하려면 볼륨과 연관된 StorageClass에서 allowVolumeExpansion을 true로 설정합니다. 영구 볼륨의 크기를 조정해야 할 때마다 영구 볼륨 클레임의 'pec.resources.requests.storage' 주석을 필요한 볼륨 크기로 편집합니다. Trident는 스토리지 클러스터의 볼륨 크기를 자동으로 조정합니다.

기존 볼륨을 **Kubernetes**로 임포트

볼륨 가져오기를 사용하면 기존 스토리지 볼륨을 Kubernetes 환경으로 가져올 수 있습니다. 이는 현재 ONTAP-NAS, ONTAP-NAS-flexgroup, solidfire-san, Azure-NetApp-files, AWS-CV가 지원하고 있습니다. 그리고 “GCP-CV”의 드라이버도 있습니다. 이 기능은 기존 애플리케이션을 Kubernetes로 포팅하거나 재해 복구 시나리오에서 유용합니다.

ONTAP 및 'solidfire-san' 드라이버를 사용하는 경우, 'tridentctl import volume <backend-name><volume-name> -f/path/PVC.YAML' 명령을 사용하여 Astra Trident에서 관리할 기존 볼륨을 Kubernetes로 가져옵니다. 볼륨 가져오기 명령에 사용되는 PVC YAML 또는 JSON 파일은 Astra Trident를 프로비저닝자로 식별하는 스토리지 클래스를 가리킵니다. NetApp HCI/SolidFire 백엔드를 사용할 경우 볼륨 이름이 고유한지 확인합니다. 볼륨 이름이 중복되면 볼륨을 고유한 이름으로 복제하여 볼륨 가져오기 기능에서 볼륨 이름을 구분할 수 있도록 합니다.

'AWS-CV', 'Azure-NetApp-files' 또는 'GCP-CV' 드라이버를 사용하는 경우, 'tridentctl import volume <backend-name><volume path> -f/path/PVC.YAML' 명령을 사용하여 Astra Trident에서 관리할 Kubernetes로 볼륨을 가져오십시오. 이렇게 하면 고유한 볼륨 참조가 보장됩니다.

위 명령을 실행하면 Astra Trident가 백엔드에서 볼륨을 찾고 해당 크기를 읽습니다. 구성된 PVC의 볼륨 크기를 자동으로 추가(필요한 경우 덮어쓰기)합니다. 그런 다음 Astra Trident가 새로운 PV를 생성하고 Kubernetes가 PVC를 PV에 결합합니다.

특정 가져온 PVC가 필요한 컨테이너를 배포한 경우 PVC/PV 쌍이 볼륨 가져오기 프로세스를 통해 바인딩될 때까지 보류 상태로 유지됩니다. PVC/PV 쌍이 바인딩되면 다른 문제가 없는 한 컨테이너가 나타나야 합니다.

OpenShift 서비스를 배포합니다

OpenShift 부가 가치 클러스터 서비스는 클러스터 관리자와 호스팅 중인 애플리케이션에 중요한 기능을 제공합니다. 이러한 서비스가 사용되는 스토리지는 노드 로컬 리소스를 사용하여 프로비저닝할 수 있지만, 이로 인해 서비스의 용량, 성능, 복구 가능성 및 지속 가능성이 제한되기도 합니다. 엔터프라이즈 스토리지 어레이를 활용하여 이러한 서비스에 필요한 용량을 제공하면 서비스를 대폭 향상시킬 수 있습니다. 그러나 모든 애플리케이션과 마찬가지로 OpenShift와 스토리지 관리자는 긴밀하게 협력하여 각 애플리케이션에 가장 적합한 옵션을 결정해야 합니다. Red Hat 문서는 요구 사항을 결정하고 사이징 및 성능 요구 사항을 충족할 수 있도록 적극 활용해야 합니다.

레지스트리 서비스

레지스트리의 스토리지 배포 및 관리는 에 설명되어 있습니다 "NetApp.IO를 참조하십시오" 에 있습니다 "블로그".

로깅 서비스

다른 OpenShift 서비스와 마찬가지로 로깅 서비스는 Ansible을 사용하여 인벤토리 파일에서 제공하는 구성 매개 변수로 배포됩니다 호스트가 플레이북에 제공됩니다. OpenShift를 설치한 후 초기 OpenShift 설치 중에 로깅을 배포하고 로깅을 배포하는 두 가지 설치 방법이 제공됩니다.

> Red Hat OpenShift 버전 3.9를 기준으로 공식 문서는 데이터 손상 관련 우려 때문에 로깅 서비스에 NFS를 사용할 것을 권장합니다. 이는 제품에 대한 Red Hat 테스트를 기반으로 합니다. ONTAP의 NFS 서버에는 이러한 문제가 없으며 로깅 구축을 쉽게 되돌릴 수 있습니다. 궁극적으로, 로깅 서비스를 위한 프로토콜을 선택할 수 있습니다. 두 가지 모두 NetApp 플랫폼을 사용할 때 효과가 있으며 원할 경우 NFS를 피할 이유가 없습니다.

로깅 서비스에서 NFS를 사용하도록 선택한 경우 설치 관리자의 실패를 방지하려면 Ansible 변수 "openshift_enable_unsupported_configurations"를 "true"로 설정해야 합니다.

시작하십시오

로깅 서비스는 필요에 따라 두 애플리케이션 및 OpenShift 클러스터 자체의 핵심 운영에 구축할 수 있습니다. 작업 로깅을 배포하려는 경우 변수 "openshift_logging_use_ops"를 "true"로 지정하면 서비스의 인스턴스 두 개가 만들어집니다. 작업에 대한 로깅 인스턴스를 제어하는 변수에는 "ops"가 포함되어 있지만 응용 프로그램의 인스턴스는 그렇지 않습니다.

기본 서비스에서 올바른 스토리지를 활용할 수 있도록 구축 방법에 따라 Ansible 변수를 구성하는 것이 중요합니다. 각 배포 방법에 대한 옵션을 살펴보겠습니다.

> 아래 표에는 로깅 서비스와 관련된 스토리지 구성과 관련된 변수만 포함되어 있습니다. 에서 다른 옵션을 찾을 수 있습니다 "RedHat OpenShift 로깅 설명서" 배포 내용에 따라 검토, 구성 및 사용해야 합니다.

아래 표의 변수는 제공된 세부 정보를 사용하여 로깅 서비스에 대한 PV 및 PVC를 생성하는 Ansible 플레이북을 만듭니다. 이 방법은 OpenShift 설치 후 구성 요소 설치 플레이북을 사용하는 것보다 훨씬 덜 유연하지만, 기존 볼륨을 사용할 수 있는 경우 옵션으로 제공됩니다.

변수	세부 정보
"openshift_logging_storage_kind"	설치 프로그램이 로깅 서비스에 대한 NFS PV를 생성하도록 'NFS'로 설정합니다.
"openshift_logging_storage_host"를 선택합니다	NFS 호스트의 호스트 이름 또는 IP 주소입니다. 이 경우 가상 머신의 데이터 LIF로 설정해야 합니다.

변수	세부 정보
'openshift_logging_storage_nfs_directory	NFS 내보내기의 마운트 경로입니다. 예를 들어 볼륨이 '/openshift_logging'으로 가정되는 경우 이 변수에 해당 경로를 사용합니다.
'openshift_logging_storage_volume_name'	생성할 PV의 이름(예: PV_ose_logs)입니다.
"openshift_logging_storage_volume_size"	NFS 내보내기의 크기(예: 100Gi)입니다.

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

변수	세부 정보
"openshift_logging_es_pvc_dynamic"	동적으로 프로비저닝된 볼륨을 사용하려면 true로 설정합니다.
'openshift_logging_es_pvc_storage_class_name'	PVC에 사용될 스토리지 클래스의 이름입니다.
"openshift_logging_es_pvc_size"를 선택합니다	PVC에서 요청된 체적의 크기입니다.
"openshift_logging_es_pvc_prefix"	로깅 서비스에서 사용하는 PVC의 접두사입니다.
"openshift_logging_es_ops_pvc_dynamic"	작업 로깅 인스턴스에 동적으로 프로비저닝된 볼륨을 사용하려면 "true"로 설정합니다.
'openshift_logging_es_ops_pvc_storage_class_name'	작업 로깅 인스턴스에 대한 스토리지 클래스의 이름입니다.
"openshift_logging_es_ops_pvc_size"를 선택합니다	작업 인스턴스에 대한 볼륨 요청의 크기입니다.
"openshift_logging_es_ops_pvc_prefix"	ops instance PVCs(ops 인스턴스 PVC)의 접두사입니다.

로깅 스택을 배포합니다

초기 OpenShift 설치 프로세스의 일부로 로깅을 배포하는 경우 표준 배포 프로세스만 따르면 됩니다. Ansible이 완료되는 즉시 서비스를 이용할 수 있도록 필요한 서비스와 OpenShift 개체를 구성 및 배포합니다.

하지만 초기 설치 후에 구축할 경우 구성 요소 플레이북을 Ansible에서 사용해야 합니다. 이 프로세스는 다른 버전의 OpenShift에서 약간 변경될 수 있으므로 반드시 읽고 따라야 합니다 "RedHat OpenShift Container Platform 3.11 설명서" 를 참조하십시오.

메트릭 서비스

메트릭 서비스는 관리자에게 OpenShift 클러스터의 상태, 리소스 활용도 및 가용성에 대한 중요한 정보를 제공합니다. 또한 POD 자동 크기 조정 기능도 필요하며, 많은 조직에서 비용 청구 및/또는 애플리케이션 표시를 위해 메트릭 서비스의 데이터를 사용합니다.

로깅 서비스 및 OpenShift와 마찬가지로 Ansible을 사용하여 메트릭 서비스를 배포합니다. 또한 로깅 서비스와 마찬가지로 메트릭 서비스는 클러스터의 초기 설정 중에 또는 구성 요소 설치 방법을 사용하여 작동 후에 배포될 수 있습니다. 다음 표에는 메트릭 서비스에 대한 영구 스토리지를 구성할 때 중요한 변수가 나와 있습니다.

> 아래 표에는 메트릭 서비스와 관련된 스토리지 구성과 관련된 변수만 포함되어 있습니다. 문서에 나와 있는 다른 많은 옵션은 배포 내용에 따라 검토, 구성 및 사용해야 합니다.

변수	세부 정보
"openshift_metrics_storage_kind"	설치 프로그램이 로깅 서비스에 대한 NFS PV를 생성하도록 'NFS'로 설정합니다.
'openshift_metrics_storage_host	NFS 호스트의 호스트 이름 또는 IP 주소입니다. SVM을 위한 데이터 LIF로 설정해야 합니다.
'openshift_metrics_storage_nfs_directory	NFS 내보내기의 마운트 경로입니다. 예를 들어, 볼륨이 '/openshift_metrics'로 가정되는 경우 이 변수에 해당 경로를 사용합니다.
'openshift_metrics_storage_volume_name'	생성할 PV의 이름(예: PV_ose_metrics).
'openshift_metrics_storage_volume_size	NFS 내보내기의 크기(예: 100Gi)입니다.

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

변수	세부 정보
'openshift_metrics_cassandra_pvc_prefix'	지표 PVC에 사용할 접두사입니다.
'openshift_metrics_cassandra_pvc_size	요청할 볼륨의 크기입니다.
'openshift_metrics_cassandra_storage_type'	메트릭에 사용할 스토리지 유형으로, 적절한 스토리지 클래스로 PVC를 생성하려면 Ansible에서 이를 동적 으로 설정해야 합니다.
'openshift_metrics_cassanda_pvc_storage_class_name'	사용할 스토리지 클래스의 이름입니다.

메트릭 서비스를 구축합니다

호스트/인벤토리 파일에 정의된 적절한 Ansible 변수를 사용하여 서비스를 구축하십시오. OpenShift 설치 시 배포하는 경우 PV가 자동으로 생성되고 사용됩니다. OpenShift를 설치한 후 구성 요소 플레이북을 사용하여 배포하는 경우, Ansible이 필요한 PVC를 만들고 Astra Trident가 PVC를 위한 스토리지를 프로비저닝하면 서비스를 배포합니다.

위의 변수와 배포 프로세스는 각 OpenShift 버전에 따라 변경될 수 있습니다. 검토 후 준수해야 합니다 "RedHat의 OpenShift 배포 가이드" 사용자 환경에 맞게 구성되도록 사용자의 버전에 대해.

데이터 보호

NetApp 스토리지 플랫폼이 제공하는 데이터 보호 및 복구 성능 옵션에 대해 알아보십시오. Astra Trident는 이러한 기능 중 일부를 활용할 수 있는 볼륨을 프로비저닝할 수 있습니다. 지속성 요구사항이 있는 각 애플리케이션에 대한 데이터 보호 및 복구 전략이 있어야 합니다.

를 백업합니다 `etcd` 클러스터 데이터

Astra Trident는 Kubernetes 클러스터의 "etcd" 데이터베이스에 메타데이터를 저장합니다. 재해 시나리오에서 Kubernetes 클러스터를 복구하려면 주기적으로 "etcd" 클러스터 데이터를 백업하는 것이 중요합니다.

단계

1. "etcctl snapshot save" 명령을 사용하면 "etcd" 클러스터의 시점 스냅샷을 만들 수 있습니다.

```
sudo docker run --rm -v /backup:/backup \
  --network host \
  -v /etc/kubernetes/pki/etcd:/etc/kubernetes/pki/etcd \
  --env ETCDCTL_API=3 \
  k8s.gcr.io/etcd-amd64:3.2.18 \
  etcdctl --endpoints=https://127.0.0.1:2379 \
  --cacert=/etc/kubernetes/pki/etcd/ca.crt \
  --cert=/etc/kubernetes/pki/etcd/healthcheck-client.crt \
  --key=/etc/kubernetes/pki/etcd/healthcheck-client.key \
  snapshot save /backup/etcd-snapshot.db
```

이 명령은 etcd 컨테이너를 스핀업하여 etcd 스냅샷을 생성하고 이를 '/backup' 디렉토리에 저장합니다.

2. 재해가 발생할 경우 etcd 스냅샷을 사용하여 Kubernetes 클러스터를 사용할 수 있습니다. "etcctl snapshot restore" 명령을 사용하여 "/var/lib/etcd" 폴더에 생성된 특정 스냅샷을 복구합니다. 복원 후 '/var/lib/etcd' 폴더가 'ember' 폴더로 채워졌는지 확인합니다. 다음은 'etcctl snapshot restore' 명령의 예입니다.

```
# etcdctl snapshot restore '/backup/etcd-snapshot-latest.db' ; mv
/default.etcd/member/ /var/lib/etcd/
```

3. Kubernetes 클러스터를 초기화하기 전에 필요한 인증서를 모두 복사합니다.
4. "--ignore-preflight-errors=DirAvailable—var-lib-etcd' 플래그를 사용하여 클러스터를 생성합니다.
5. 클러스터가 완료되면 kubbe-system 포드가 시작되었는지 확인합니다.
6. Trident에서 만든 사용자 지정 리소스가 있는지 확인하고 Trident 개체를 검색하여 모든 데이터를 사용할 수 있는지 확인하려면 "kubbeck get CRD" 명령을 사용합니다.

ONTAP 스냅샷을 사용하여 날짜를 복구합니다

스냅샷은 애플리케이션 데이터에 대한 시점 복구 옵션을 제공하여 중요한 역할을 합니다. 그러나 스냅샷은 자체적으로 백업되는 것이 아니며, 스토리지 시스템 장애나 기타 재난으로부터 데이터를 보호하지 않습니다. 그러나 대부분의 경우 데이터를 쉽고 빠르고 쉽게 복구할 수 있습니다. ONTAP 스냅샷 기술을 사용하여 볼륨을 백업하는 방법과 복원하는 방법에 대해 알아보십시오.

- 백엔드에 스냅샷 정책이 정의되지 않은 경우 기본적으로 "없음" 정책을 사용합니다. 이로 인해 ONTAP에서 자동 스냅샷이 생성되지 않습니다. 그러나 스토리지 관리자는 ONTAP 관리 인터페이스를 통해 수동 스냅샷을 생성하거나 스냅샷 정책을 변경할 수 있습니다. Trident 작업에는 영향을 주지 않습니다.
- 스냅샷 디렉토리는 기본적으로 숨겨져 있습니다. 이를 통해 ONTAP-NAS와 ONTAP-NAS-이코노미 드라이버를 사용하여 프로비저닝된 볼륨의 호환성을 극대화할 수 있습니다. ONTAP-NAS 및 ONTAP-NAS-이코노미 드라이버를 사용할 경우 애플리케이션에서 스냅샷의 데이터를 직접 복구할 수 있도록 `.snapshot' 디렉터리를 활성화합니다.
- 'volume snapshot restore' ONTAP CLI 명령을 사용하여 이전 스냅샷에 기록된 상태로 볼륨을 복원합니다. 스냅샷 복사본을 복구할 때 복구 작업은 기존 볼륨 구성을 덮어씁니다. 스냅샷 복사본이 생성된 후 볼륨의 데이터에 대한 모든 변경 사항은 손실됩니다.

```
cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```

ONTAP를 사용하여 데이터 복제

스토리지 시스템 장애로 인한 데이터 손실로부터 데이터를 복제하는 것은 중요한 역할을 할 수 있습니다.

> ONTAP 복제 기술에 대한 자세한 내용은 를 참조하십시오 "ONTAP 설명서".

SnapMirror SVM(Storage Virtual Machines) 복제

을 사용할 수 있습니다 "SnapMirror를 참조하십시오" 전체 SVM을 복제하며, 여기에는 구성 설정 및 볼륨이 포함됩니다. 재해가 발생할 경우 SnapMirror 대상 SVM을 활성화하여 데이터 제공을 시작할 수 있습니다. 시스템이 복원되면 기본 시스템으로 다시 전환할 수 있습니다.

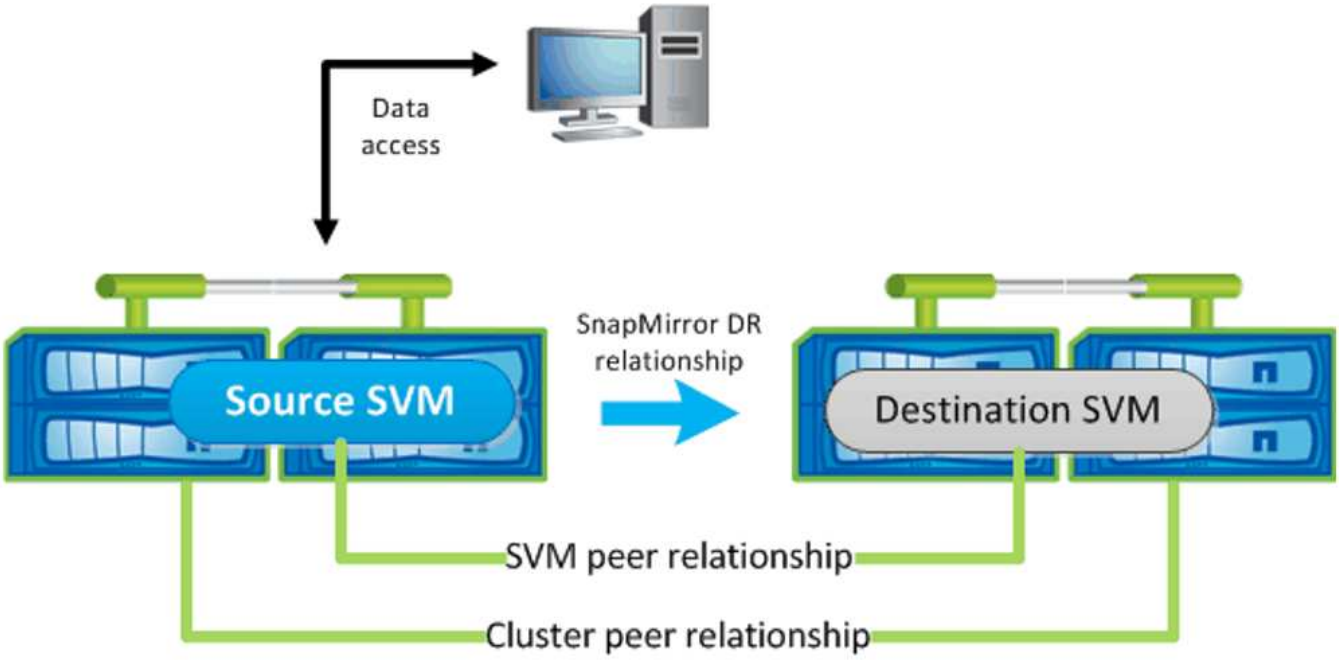
Astra Trident는 복제 관계 자체를 구성할 수 없기 때문에 스토리지 관리자는 ONTAP의 SnapMirror SVM 복제 기능을 사용하여 DR(재해 복구) 대상에 볼륨을 자동으로 복제할 수 있습니다.

SnapMirror SVM 복제 기능을 사용할 계획이거나 현재 기능을 사용 중인 경우 다음을 고려하십시오.

- SVM-DR이 활성화된 각 SVM에 대해 별개의 백엔드를 생성해야 합니다.
- 필요한 경우를 제외하고 복제된 백엔드를 선택하지 않도록 스토리지 클래스를 구성해야 합니다. 이는 SVM-DR을 지원하는 백엔드에 복제 관계를 프로비저닝하지 않아도 되는 볼륨이 생기지 않도록 하는 데 중요합니다.
- 애플리케이션 관리자는 데이터 복제와 관련된 추가 비용 및 복잡성을 이해하고 데이터 복제를 활용하기 전에 복구 계획을 결정해야 합니다.
- SnapMirror 대상 SVM을 활성화하기 전에 예약된 SnapMirror 전송을 모두 중지하고, 진행 중인 SnapMirror 전송을 모두 중단하고, 복제 관계를 중지하고, 소스 SVM을 중지한 다음, SnapMirror 대상 SVM을 시작하십시오.
- Astra Trident는 SVM 장애를 자동으로 감지하지 않습니다. 따라서 장애가 발생하면 관리자는 'tridentctl backend update' 명령을 실행하여 Trident가 새 백엔드로 장애 조치를 트리거해야 합니다.

다음은 SVM 설정 단계에 대한 개요입니다.

- 소스 클러스터와 타겟 클러스터 및 SVM 간 피어링을 설정합니다.
- '-subtype DP-destination' 옵션을 사용하여 대상 SVM을 생성합니다.
- 필요한 간격으로 복제가 수행되도록 복제 작업 스케줄을 생성합니다.
- 소스 SVM 구성과 소스 SVM 인터페이스가 타겟으로 복제되도록 '-identity-preserve true' 옵션을 사용하여 타겟 SVM에서 소스 SVM으로 SnapMirror 복제를 생성합니다. 대상 SVM에서 SnapMirror SVM 복제 관계를 초기화합니다.



Trident를 위한 재해 복구 워크플로우

Astra Trident 19.07 이상 Kubernetes CRD를 사용하여 자체 상태를 저장 및 관리합니다. 이 경우 Kubernetes 클러스터의 etcd를 사용하여 메타데이터를 저장합니다. 여기에서는 Kubernetes "etcd" 데이터 파일과 인증서가 NetApp FlexVolume에 저장되어 있다고 가정합니다. 이 FlexVolume은 SVM에 상주하며 보조 사이트의 대상 SVM과 SnapMirror SVM-DR 관계가 있습니다.

다음 단계에서는 재해 발생 시 Astra Trident를 사용하여 단일 마스터 Kubernetes 클러스터를 복구하는 방법을 설명합니다.

1. 소스 SVM에 장애가 발생하면 SnapMirror 타겟 SVM을 활성화합니다. 이렇게 하려면 예약된 SnapMirror 전송을 중지하고, 지속적인 SnapMirror 전송을 중단하고, 복제 관계를 중단하고, 소스 SVM을 중지하고, 타겟 SVM을 시작해야 합니다.
2. 대상 SVM에서 Kubernetes "etcd" 데이터 파일 및 인증서가 포함된 볼륨을 마스터 노드로 설정할 호스트에 마운트합니다.
3. /etc/Kubernetes/pki 아래에 있는 Kubernetes 클러스터와 관련된 모든 필수 인증서를 복사하고, '/var/lib/etcd' 아래에 있는 etcd member 파일을 복사합니다.
4. '--ignore-preflight-errors=DirAvailable—var-lib-etcd' 플래그를 사용하여 kubeadm init 명령을 사용하여 Kubernetes 클러스터를 생성합니다. Kubernetes 노드에 사용되는 호스트 이름은 소스 Kubernetes 클러스터와 동일해야 합니다.
5. 'kubeck get CRD' 명령을 실행하여 모든 Trident 사용자 지정 리소스가 표시되는지 확인하고 Trident 객체를 검색하여 모든 데이터를 사용할 수 있는지 확인합니다.
6. './tridentctl update backend <backend-name> -f <backend-json-file> -n <namespace>' 명령을 실행하여 새 대상 SVM 이름을 반영하도록 필요한 모든 백엔드를 업데이트합니다.

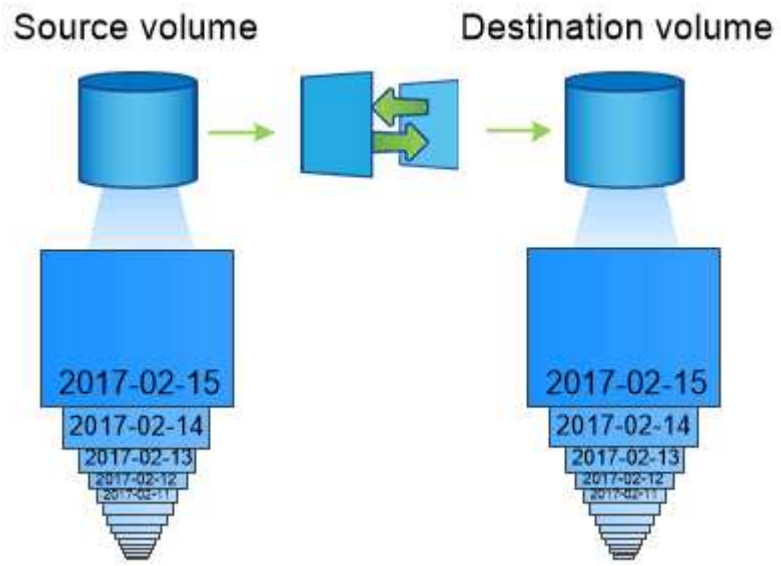
> 애플리케이션의 영구 볼륨의 경우, 대상 SVM이 활성화될 때 Trident가 프로비저닝한 모든 볼륨이 데이터 제공을 시작합니다. 위에서 설명한 단계를 사용하여 대상 측에 Kubernetes 클러스터를 설정한 후에는 모든 구축과 포드가 시작되고 패키지 애플리케이션은 문제 없이 실행되어야 합니다.

SnapMirror 볼륨 복제

ONTAP SnapMirror 볼륨 복제는 재해 복구 기능으로, 볼륨 레벨의 운영 스토리지에서 대상 스토리지로 페일오버할 수 있도록 지원합니다. SnapMirror는 스냅샷을 동기화하여 보조 스토리지에 운영 스토리지의 볼륨 복제본 또는 미러를 생성합니다.

다음은 ONTAP SnapMirror 볼륨 복제 설정 단계에 대한 개요입니다.

- 볼륨이 상주하는 클러스터와 볼륨의 데이터를 제공하는 SVM 간에 피어링을 설정합니다.
- 관계의 동작을 제어하고 해당 관계에 대한 구성 특성을 지정하는 SnapMirror 정책을 생성합니다.
- 를 사용하여 타겟 볼륨과 소스 볼륨 사이에 SnapMirror 관계를 생성합니다 "d9934e78a9254dde4a227181c30fa2d2" 적절한 SnapMirror 정책을 할당합니다.
- SnapMirror 관계가 생성된 후 소스 볼륨에서 타겟 볼륨으로의 기본 전송이 완료되도록 관계를 초기화합니다.



Trident를 위한 **SnapMirror** 볼륨 재해 복구 워크플로우

다음 단계에서는 Astra Trident를 사용하여 단일 마스터 Kubernetes 클러스터를 복구하는 방법을 설명합니다.

1. 재해가 발생할 경우 예약된 SnapMirror 전송을 모두 중지하고 진행 중인 SnapMirror 전송을 모두 중단하십시오. 대상 볼륨이 읽기/쓰기가 되도록 대상 볼륨과 소스 볼륨 간의 복제 관계를 중단하십시오.
2. 대상 SVM에서 Kubernetes "etcd" 데이터 파일 및 인증서가 포함된 볼륨을 호스트에 마운트하고 마스터 노드로 설정됩니다.
3. /etc/Kubernetes/pki 아래에 있는 Kubernetes 클러스터와 관련된 모든 필수 인증서를 복사하고, '/var/lib/etcd' 아래에 있는 etcd member 파일을 복사합니다.
4. '--ignore-preflight-errors=DirAvailable—var-lib-etcd' 플래그를 사용하여 "kubeadm init" 명령을 실행하여 Kubernetes 클러스터를 생성합니다. 호스트 이름은 소스 Kubernetes 클러스터와 같아야 합니다.
5. 'kubeck get CRD' 명령을 실행하여 모든 Trident 사용자 지정 리소스가 검색되었는지 확인하고 Trident 객체를 검색하여 모든 데이터를 사용할 수 있는지 확인합니다.
6. 이전 백엔드를 정리하고 Trident에 새 백엔드를 만듭니다. 새로운 관리 및 데이터 LIF, 새로운 SVM 이름 및 대상 SVM의 암호를 지정합니다.

다음 단계에서는 재해 발생 시 컨테이너화된 워크로드에 SnapMirror 대상 볼륨을 제공하는 방법을 설명합니다.

1. 예약된 모든 SnapMirror 전송을 중지하고 진행 중인 모든 SnapMirror 전송을 중단합니다. 대상 볼륨이 읽기/쓰기가 되도록 대상 볼륨과 소스 볼륨 간의 복제 관계를 중단하십시오. 소스 SVM의 볼륨에 연결된 PVC를 사용하는 구축을 정리합니다.
2. 위에서 설명한 단계를 사용하여 대상 측에 Kubernetes 클러스터를 설정한 후 Kubernetes 클러스터에서 배포, PVC 및 PV를 정리합니다.
3. Trident에서 새로운 관리 및 데이터 LIF, 새 SVM 이름 및 대상 SVM의 암호를 지정하여 새 백엔드를 생성합니다.
4. Trident 가져오기 기능을 사용하여 새 PVC에 바인딩된 PV로 필요한 볼륨을 가져옵니다.
5. 새로 생성된 PVC와 함께 애플리케이션 배포를 재배포합니다.

Element 스냅샷을 사용하여 데이터 복구

볼륨에 대한 스냅샷 스케줄을 설정하고 필요한 간격으로 스냅샷을 생성하도록 하여 Element 볼륨의 데이터를 백업합니다. Element UI 또는 API를 사용하여 스냅샷 스케줄을 설정해야 합니다. 현재 '솔드파이어-SAN' 드라이버를 통해 스냅샷 스케줄을 볼륨으로 설정할 수 없습니다.

데이터가 손상된 경우 Element UI 또는 API를 사용하여 특정 스냅샷을 선택하고 볼륨을 스냅숏으로 수동으로 롤백할 수 있습니다. 이렇게 하면 스냅샷이 생성된 이후 볼륨에 대한 모든 변경 사항이 복구됩니다.

보안

여기에 나열된 권장 사항을 사용하여 Astra Trident 설치가 안전한지 확인합니다.

자체 네임스페이스에서 Astra Trident를 실행합니다

애플리케이션, 애플리케이션 관리자, 사용자 및 관리 애플리케이션에서 Astra Trident 객체 정의 또는 Pod에 액세스하여 안정적인 스토리지를 보장하고 잠재적인 악성 활동을 차단하는 것이 중요합니다.

다른 애플리케이션과 사용자를 Astra Trident에서 분리하려면 항상 고유한 Kubernetes 네임스페이스('트리덴트')에 Astra Trident를 설치하십시오. Astra Trident를 자체 네임스페이스에 두면 Kubernetes 관리 담당자만 Astra Trident POD와 이름이 같은 CRD 객체에 저장된 아티팩트(예: 백엔드 및 CHAP 암호)에 액세스할 수 있습니다. 관리자만이 Astra Trident 네임스페이스에 액세스할 수 있도록 하고 "tridentctl" 응용 프로그램에 액세스할 수 있도록 해야 합니다.

ONTAP SAN 백엔드에 CHAP 인증을 사용합니다

Astra Trident는 ONTAP SAN 워크로드에 대한 CHAP 기반 인증(ONTAP-SAN과 ONTAP-SAN 절약 드라이버 사용)을 지원합니다. 호스트와 스토리지 백엔드 간의 인증을 위해 Astra Trident와 양방향 CHAP를 사용하는 것이 좋습니다.

SAN 스토리지 드라이버를 사용하는 ONTAP 백엔드의 경우 Astra Trident는 양방향 CHAP를 설정하고 "tridentctl"을 통해 CHAP 사용자 이름과 암호를 관리할 수 있습니다. 을 참조하십시오 "여기" Astra Trident가 ONTAP 백엔드에서 CHAP를 구성하는 방법을 이해합니다.

> ONTAP 백엔드에 대한 CHAP 지원은 Trident 20.04 이상에서 사용할 수 있습니다.

NetApp HCI 및 SolidFire 백엔드에서 CHAP 인증을 사용합니다

양방향 CHAP를 구축하여 호스트와 NetApp HCI 및 SolidFire 백엔드 간의 인증을 보장하는 것이 좋습니다. Astra Trident는 테넌트당 2개의 CHAP 암호를 포함하는 비밀 객체를 사용합니다. Trident를 CSI 구축 담당자로 설치하면 CHAP 암호를 관리하고 해당 PV의 "tridentvolume" CR 객체로 저장합니다. PV를 생성할 때 CSI Astra Trident는 CHAP 암호를 사용하여 iSCSI 세션을 시작하고 CHAP를 통해 NetApp HCI 및 SolidFire 시스템과 통신합니다.

> CSI Trident에서 생성한 볼륨은 볼륨 액세스 그룹과 연결되지 않습니다.

CSI가 아닌 프런트엔드에서는 작업자 노드의 디바이스로 볼륨을 연결하는 작업을 Kubernetes에서 처리합니다. 볼륨 생성 후 Astra Trident는 NetApp HCI/SolidFire 시스템에 API 호출을 통해 해당 테넌트의 암호가 아직 없는 경우 비밀을 검색합니다. 그런 다음 Astra Trident가 Kubernetes에 비밀을 전달합니다. 각 노드에 위치한 kubelet은 Kubernetes API를 통해 기밀에 액세스하고 이를 사용하여 볼륨에 액세스하는 각 노드와 볼륨이 있는 NetApp HCI/SolidFire 시스템 간에 CHAP를 실행/사용하도록 설정합니다.

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이센스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이센스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이센스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 http://www.netapp.com/TM에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.