



Oracle 데이터베이스 마이그레이션

Enterprise applications

NetApp
May 09, 2024

목차

Oracle 데이터베이스 마이그레이션	1
Oracle 데이터베이스를 ONTAP 스토리지 시스템으로 마이그레이션	1
Oracle 데이터베이스 마이그레이션 계획	1
절차를 참조하십시오	4
Oracle 마이그레이션 절차 샘플 스크립트	104

Oracle 데이터베이스 마이그레이션

Oracle 데이터베이스를 ONTAP 스토리지 시스템으로 마이그레이션

새로운 스토리지 플랫폼의 기능 활용에는 한 가지 필연적인 요구사항이 있으므로 데이터를 새로운 스토리지 시스템에 배치해야 합니다. ONTAP를 사용하면 ONTAP에서 ONTAP로의 마이그레이션 및 업그레이드, 외부 LUN 가져오기, 호스트 운영 체제 또는 Oracle 데이터베이스 소프트웨어를 직접 사용하는 절차 등 마이그레이션 프로세스를 간소화할 수 있습니다.



이 문서는 이전에 게시된 기술 보고서 _TR-4534: Oracle 데이터베이스를 NetApp 스토리지 시스템으로 마이그레이션 _을(를) 대체합니다

새 데이터베이스 프로젝트의 경우 데이터베이스 및 애플리케이션 환경이 제대로 구축되기 때문에 이 문제가 발생하지 않습니다. 그러나 마이그레이션은 비즈니스 중단, 마이그레이션 완료에 필요한 시간, 필요한 기술 세트 및 위험 최소화와 관련하여 특별한 과제를 안고 있습니다.

스크립트

샘플 스크립트는 이 설명서에 나와 있습니다. 이러한 스크립트는 사용자 오류 가능성을 줄이기 위해 마이그레이션의 다양한 측면을 자동화하는 샘플 방법을 제공합니다. 이 스크립트는 마이그레이션을 담당하는 IT 직원의 전반적인 요구사항을 줄이고 전반적인 프로세스를 가속화할 수 있습니다. 이 스크립트는 NetApp 프로페셔널 서비스 및 NetApp 파트너가 수행하는 실제 마이그레이션 프로젝트로부터 작성되었습니다. 이 설명서 전반에 걸쳐 사용 예제가 나와 있습니다.

Oracle 데이터베이스 마이그레이션 계획

Oracle 데이터 마이그레이션은 데이터베이스, 호스트 또는 스토리지 배열의 세 가지 레벨 중 하나에서 수행할 수 있습니다.

데이터베이스, 호스트 운영 체제 또는 스토리지 시스템 등 전체 솔루션의 어떤 구성 요소가 데이터 이동을 담당하는지에 따라 차이가 있습니다.

아래 그림에서는 마이그레이션 수준과 데이터 흐름의 예를 보여 줍니다. 데이터베이스 레벨 마이그레이션의 경우 데이터가 원래 스토리지 시스템에서 호스트 및 데이터베이스 계층을 통해 새로운 환경으로 이동됩니다. 호스트 레벨 마이그레이션은 비슷하지만 데이터가 애플리케이션 계층을 거치지 않고 호스트 프로세스를 사용하여 새 위치에 기록됩니다. 마지막으로, 스토리지 레벨 마이그레이션을 통해 NetApp FAS 시스템과 같은 어레이가 데이터 이동을 담당합니다.

[오류: 그래픽 이미지가 없습니다]

데이터베이스 수준 마이그레이션은 일반적으로 Oracle 계층에서 마이그레이션을 완료하기 위해 대기 데이터베이스를 통한 Oracle 로그 전달을 사용하는 것을 의미합니다. 호스트 레벨 마이그레이션은 호스트 운영 체제 구성의 기본 기능을 사용하여 수행됩니다. 이 구성에는 cp, tar 및 Oracle RMAN(Recovery Manager)과 같은 명령을 사용하거나 LVM(Logical Volume Manager)을 사용하여 파일 시스템의 기본 바이트를 재배치하는 파일 복사 작업이 포함됩니다. Oracle ASM(Automatic Storage Management)은 데이터베이스 애플리케이션 수준 이하로 실행되기 때문에 호스트 수준 기능으로 분류됩니다. ASM은 호스트에서 일반적인 논리적 볼륨 관리자를 대체합니다. 마지막으로, 데이터는 스토리지 어레이 레벨에서 마이그레이션될 수 있으며, 이는 운영 체제 레벨 아래에서 마이그레이션됩니다.

계획 고려 사항

마이그레이션을 위한 최상의 옵션은 마이그레이션할 환경의 규모, 다운타임을 방지해야 하는 필요성, 마이그레이션을 수행하는 데 필요한 전반적인 노력 등 여러 요소의 조합에 따라 달라집니다. 대규모 데이터베이스에는 마이그레이션을 위해 더 많은 시간과 노력이 필요하지만, 이와 같은 마이그레이션의 복잡성은 최소화됩니다. 작은 데이터베이스는 신속하게 마이그레이션할 수 있지만, 수천 개의 마이그레이션을 해야 하는 경우 그 규모가 커지면 복잡한 문제가 발생할 수 있습니다. 마지막으로, 데이터베이스가 클수록 비즈니스 크리티컬에 해당할 가능성이 커지므로 백엔드 경로를 유지하면서 다운타임을 최소화해야 합니다.

여기에서는 마이그레이션 전략 계획을 위한 몇 가지 고려 사항에 대해 설명합니다.

데이터 크기입니다

크기가 컷오버 시간에 영향을 미칠 필요는 없지만 마이그레이션할 데이터베이스의 크기는 마이그레이션 계획에 분명히 영향을 미칩니다. 대량의 데이터를 마이그레이션해야 하는 경우 주요 고려 사항은 대역폭입니다. 복제 작업은 일반적으로 효율적인 순차적 I/O를 사용하여 수행됩니다 보수적인 추정치로, 복제 작업에 사용 가능한 네트워크 대역폭의 50%를 사용한다고 가정합니다. 예를 들어, 8GB FC 포트는 이론적으로 약 800Mbps를 전송할 수 있습니다. 50%의 활용률을 가정하면 약 400Mbps의 속도로 데이터베이스를 복사할 수 있습니다. 따라서 10TB 데이터베이스를 이 속도로 약 7시간 내에 복사할 수 있습니다.

장거리 마이그레이션을 위해서는 일반적으로 에 설명된 로그 전달 프로세스와 같이 보다 창의적인 접근 방식이 필요합니다 "온라인 데이터 파일 이동". 장거리 IP 네트워크는 LAN 또는 SAN 속도와 가까운 곳에서는 거의 대역폭을 가지지 않습니다. 한 예로, NetApp는 아카이브 로그 생성 속도가 매우 높은 220TB 데이터베이스의 장거리 마이그레이션을 지원했습니다. 이 방법은 가능한 최대 대역폭을 제공하기 때문에 데이터 전송을 위해 선택한 방식은 테이프를 매일 배송하는 방식이었습니다.

데이터베이스 수입니다

많은 경우 대량의 데이터를 이동할 때 발생하는 문제는 데이터 크기가 아니라 데이터베이스를 지원하는 구성의 복잡성입니다. 단순히 50TB의 데이터베이스를 마이그레이션해야 한다는 사실을 아는 것만으로는 충분하지 않습니다. 단일 50TB 미션 크리티컬 데이터베이스, 4천 개의 기존 데이터베이스 컬렉션 또는 운영 데이터와 비운영 데이터의 조합이 될 수 있습니다. 경우에 따라 대부분의 데이터는 소스 데이터베이스의 클론으로 구성됩니다. 특히, 새 아키텍처에서 NetApp FlexClone 볼륨을 활용하도록 설계된 경우 이러한 클론을 쉽게 다시 생성할 수 있으므로 이러한 클론을 마이그레이션할 필요가 없습니다.

마이그레이션 계획을 수립하려면 범위에 포함되는 데이터베이스의 수와 우선 순위를 어떻게 지정해야 하는지 파악해야 합니다. 데이터베이스 수가 증가함에 따라 기본 마이그레이션 옵션은 스택에서 더 낮거나 낮은 경향이 있습니다. 예를 들어, RMAN과 짧은 운영 중단으로 단일 데이터베이스를 쉽게 복사할 수 있습니다. 이것이 호스트 수준 복제입니다.

데이터베이스가 50개인 경우 RMAN 복제본을 수신하도록 새 파일 시스템 구조를 설정하는 대신 데이터를 제자리로 이동하는 것이 더 쉬울 수 있습니다. 이 프로세스는 호스트 기반 LVM 마이그레이션을 통해 데이터를 이전 LUN에서 새 LUN으로 재배치하는 방법으로 수행할 수 있습니다. 이렇게 하면 DBA(데이터베이스 관리자) 팀에서 OS 팀으로 역할이 이전되고, 결과적으로 데이터가 데이터베이스와 관련하여 투명하게 마이그레이션됩니다. 파일 시스템 구성이 변경되지 않았습니다.

끝으로, 서버 200대에서 500개의 데이터베이스를 마이그레이션해야 하는 경우 ONTAP FLI(Foreign LUN Import) 기능과 같은 스토리지 기반 옵션을 사용하여 LUN을 직접 마이그레이션할 수 있습니다.

재건축 요구 사항

일반적으로 새 스토리지 어레이의 기능을 활용하기 위해 데이터베이스 파일 레이아웃을 변경해야 하지만 항상 그렇지는 않습니다. 예를 들어, EF-Series All-Flash 어레이의 기능은 주로 SAN 성능과 SAN 안정성을 최우선으로 합니다.

대부분의 경우 데이터 레이아웃과 관련하여 특별한 고려 사항 없이 데이터베이스를 EF-Series 어레이로 마이그레이션할 수 있습니다. 높은 IOPS, 짧은 지연 시간 및 강력한 안정성만 필요합니다. RAID 구성 또는 Dynamic Disk Pool과 같은 요소와 관련된 모범 사례가 있지만 EF-Series 프로젝트에서 이러한 기능을 활용하기 위해 전체 스토리지 아키텍처를 크게 변경할 필요는 없습니다.

반면, ONTAP로 마이그레이션하려면 일반적으로 최종 구성이 최대한의 가치를 제공할 수 있도록 데이터베이스 레이아웃을 좀 더 고려해야 합니다. ONTAP는 그 자체로 특별한 아키텍처 노력 없이 데이터베이스 환경에 많은 기능을 제공합니다. 가장 중요한 것은 현재 하드웨어의 수명이 다할 때 새 하드웨어로 중단 없이 마이그레이션할 수 있는 기능을 제공한다는 것입니다. 일반적으로 ONTAP로의 마이그레이션은 수행해야 하는 마지막 마이그레이션입니다. 후속 하드웨어가 업그레이드되고 데이터가 중단 없이 새로운 미디어로 마이그레이션됩니다.

어떤 계획을 세우면 훨씬 더 많은 혜택을 누릴 수 있습니다. 가장 중요한 고려 사항은 스냅샷 사용과 관련됩니다. 스냅샷은 거의 즉각적인 백업, 복원 및 클론 복제 작업을 수행하기 위한 기반입니다. 스냅샷 성능의 예로, 알려진 가장 큰 용도는 6개의 컨트롤러에서 약 250개의 LUN에서 실행되는 996TB의 단일 데이터베이스를 사용하는 것입니다. 이 데이터베이스는 2분 내에 백업되고 2분 내에 복원되며 15분 내에 복제될 수 있습니다. 추가 이점으로는 워크로드의 변화에 대응하여 클러스터 주변으로 데이터를 이동하는 기능과 QoS(서비스 품질) 제어 애플리케이션을 통해 다중 데이터베이스 환경에서 양호하고 일관된 성능을 제공하는 기능이 있습니다.

QoS 제어, 데이터 재배치, 스냅샷, 클론 복제 등의 기술은 거의 모든 구성에서 작동합니다. 그러나 일반적으로 이점을 극대화하기 위해서는 몇 가지 생각이 필요합니다. 새로운 스토리지 어레이에 대한 투자를 최대화하기 위해 데이터베이스 스토리지 레이아웃을 변경해야 하는 경우도 있습니다. 호스트 기반 또는 스토리지 기반 마이그레이션은 원본 데이터 레이아웃을 복제하므로 이러한 설계 변경은 마이그레이션 전략에 영향을 미칠 수 있습니다. 마이그레이션을 완료하고 ONTAP에 최적화된 데이터 레이아웃을 제공하기 위해 추가 단계가 필요할 수 있습니다. 에 나와 있는 절차 "[Oracle 마이그레이션 절차 개요](#)" 그리고 나중에 데이터베이스를 마이그레이션하는 것뿐만 아니라 최소한의 노력으로 최적의 최종 레이아웃으로 마이그레이션하는 몇 가지 방법을 보여 줍니다.

컷오버 시간

컷오버 중에 허용되는 최대 서비스 중단 시간을 결정해야 합니다. 전체 마이그레이션 프로세스로 인해 운영이 중단된다고 생각하는 일반적인 실수입니다. 서비스 중단이 시작되기 전에 다양한 작업을 완료할 수 있으며, 다양한 옵션을 통해 운영 중단 또는 운영 중단 없이 마이그레이션을 완료할 수 있습니다. 불가피한 운영 중단이 불가피한 경우에도 컷오버 시간의 기간이 절차마다 다르므로 허용되는 최대 서비스 중단 시간을 정의해야 합니다.

예를 들어 10TB 데이터베이스를 복사하려면 일반적으로 약 7시간이 소요됩니다. 비즈니스 요구 사항이 7시간 동안 중단되는 경우 파일 복사는 쉽고 안전한 마이그레이션 옵션입니다. 5시간이 허용되지 않는 경우 간단한 로그 전달 프로세스를 수행합니다(참조 "[Oracle 로그 전달](#)")를 최소한의 노력으로 설정하여 컷오버 시간을 약 15분으로 단축할 수 있습니다. 이 시간 동안 데이터베이스 관리자가 이 프로세스를 완료할 수 있습니다. 15분이 허용되는 경우 스크리핑을 통해 최종 컷오버 프로세스를 자동화하여 컷오버 시간을 단 몇 분으로 단축할 수 있습니다. 언제든지 마이그레이션의 속도를 높일 수 있지만 시간과 노력을 들여야 합니다. 컷오버 시간 목표는 비즈니스에 허용되는 성과를 기준으로 해야 합니다.

뒤로 이동 경로

마이그레이션 시 위험을 완전히 없앨 수는 없습니다. 기술이 완벽하게 작동하더라도 사용자 오류가 발생할 가능성이 항상 있습니다. 선택한 마이그레이션 경로와 관련된 위험은 실패한 마이그레이션의 결과와 함께 고려해야 합니다. 예를 들어, Oracle ASM의 투명한 온라인 스토리지 마이그레이션 기능은 주요 기능 중 하나이며 이 방법은 가장 신뢰할 수 있는 기능 중 하나입니다. 그러나 이 메서드를 사용하여 데이터를 복구할 수 없는 방식으로 복사하고 있습니다. 드문 경우지만 ASM에서 문제가 발생하는 경우에는 쉬운 백아웃 경로가 없습니다. 유일한 옵션은 원래 환경을 복원하거나 ASM을 사용하여 마이그레이션을 원래 LUN으로 되돌리는 것입니다. 시스템에서 이러한 작업을 수행할 수 있다고 가정하면 원래 스토리지 시스템에서 스냅샷 유형 백업을 수행하면 위험이 최소화될 수 있지만 제거되지는 않습니다.

예행 연습

일부 마이그레이션 절차는 실행 전에 완전히 검증되어야 합니다. 마이그레이션 및 전환 프로세스의 예행 연습은 마이그레이션을 성공적으로 수행하고 다운타임을 최소화해야 하는 미션 크리티컬 데이터베이스에 대한 일반적인 요청입니다. 또한 사용자 수용 테스트는 마이그레이션 후 작업의 일부로 포함되는 경우가 많으며 이러한 테스트가 완료된 후에만 전체 시스템을 운영 환경으로 되돌릴 수 있습니다.

예행 연습이 필요한 경우 몇 가지 ONTAP 기능을 통해 프로세스를 훨씬 쉽게 수행할 수 있습니다. 특히 스냅샷은 테스트 환경을 재설정하고 데이터베이스 환경의 공간 효율적인 여러 복제본을 신속하게 생성할 수 있습니다.

절차를 참조하십시오

Oracle 마이그레이션 절차 개요

Oracle 마이그레이션 데이터베이스에 다양한 절차를 사용할 수 있습니다. 올바른 운영 체제는 비즈니스 요구 사항에 달려 있습니다.

대부분의 경우 시스템 관리자와 DBA는 물리적 볼륨 데이터를 재배포하거나, 미러링 및 방해 요소를 제거하거나, Oracle RMAN을 활용하여 데이터를 복제하는 방법을 선호하고 있습니다.

이러한 절차는 사용 가능한 옵션 중 일부에 익숙하지 않은 IT 직원을 위한 지침으로 주로 제공됩니다. 또한 절차를 통해 각 마이그레이션 방법에 대한 작업, 시간 요구사항 및 기술 집합 요구를 파악할 수 있습니다. 이를 통해 NetApp 및 파트너 전문 서비스 또는 IT 관리 부서와 같은 타사는 각 절차의 요구사항을 보다 완벽하게 이해할 수 있습니다.

마이그레이션 전략을 만드는 단일 모범 사례는 없습니다. 계획을 작성하려면 먼저 가용성 옵션을 파악한 다음 비즈니스 요구에 가장 적합한 방법을 선택해야 합니다. 아래 그림은 기본적인 고려 사항과 고객이 내린 일반적인 결론을 보여 주지만 모든 상황에 보편적으로 적용되는 것은 아닙니다.

예를 들어 한 단계만 거치면 전체 데이터베이스 크기에 대한 문제가 발생합니다. 다음 단계는 데이터베이스가 1TB 이하인지 여부에 따라 달라집니다. 권장 단계는 일반적인 고객 사례를 기반으로 한 권장 사항입니다. 대부분의 고객은 DataGuard를 사용하여 작은 데이터베이스를 복사하지 않을 수도 있지만 일부 고객은 그렇지 않을 수도 있습니다. 대부분의 고객은 시간이 필요하기 때문에 50TB 데이터베이스를 복사하지 않으려고 하지만 어떤 고객은 이 작업을 허용할 만큼 충분한 유지 관리 기간이 있을 수 있습니다.

가장 적합한 마이그레이션 경로에 대한 고려 사항 유형의 순서도를 찾을 수 있습니다 ["여기"](#).

온라인 데이터 파일 이동

Oracle 12cR1 이상에는 데이터베이스가 온라인 상태일 때 데이터 파일을 이동하는 기능이 포함되어 있습니다. 또한 서로 다른 파일 시스템 유형 간에 작동합니다. 예를 들어 데이터 파일을 xfs 파일 시스템에서 ASM으로 재배포할 수 있습니다. 이 방법은 필요할 수 있는 개별 데이터 파일 이동 작업의 수로 인해 일반적으로 대규모로 사용되지 않지만 데이터 파일이 적은 작은 데이터베이스에서는 고려할 만한 옵션입니다.

또한 데이터 파일을 단순히 이동하는 것이 기존 데이터베이스의 부분을 마이그레이션하는 좋은 방법입니다. 예를 들어, 유틸 블록을 오브젝트 저장소에 저장할 수 있는 FabricPool 볼륨과 같이 사용량이 적은 데이터 파일을 더 비용 효율적인 스토리지로 재배포할 수 있습니다.

데이터베이스 레벨 마이그레이션

데이터베이스 수준에서 마이그레이션하면 데이터베이스가 데이터를 재배포할 수 있습니다. 특히 로그 전송을 의미합니다. RMAN 및 ASM 같은 기술은 Oracle 제품이지만, 마이그레이션을 위해 파일을 복제하고 볼륨을 관리하는

호스트 레벨에서 작동합니다.

로그 전달

데이터베이스 수준 마이그레이션의 기반은 Oracle 아카이브 로그이며, 여기에는 데이터베이스의 변경 사항 로그가 포함됩니다. 대부분의 경우 아카이브 로그는 백업 및 복구 전략의 일부입니다. 복구 프로세스는 데이터베이스 복원으로 시작한 다음 하나 이상의 아카이브 로그를 재생하여 데이터베이스를 원하는 상태로 만듭니다. 이와 동일한 기본 기술을 사용하여 운영 중단이 거의 또는 전혀 없는 마이그레이션을 수행할 수 있습니다. 더욱 중요한 점은 이 기술을 통해 원래 데이터베이스를 그대로 유지하면서 마이그레이션을 수행할 수 있으므로 백업 경로가 유지됩니다.

마이그레이션 프로세스는 데이터베이스 백업을 보조 서버로 복원하는 것부터 시작됩니다. 다양한 방법으로 그렇게 할 수 있지만 대부분의 고객은 일반 백업 애플리케이션을 사용하여 데이터 파일을 복원합니다. 데이터 파일이 복원되면 사용자가 로그 전달 방법을 설정합니다. 기본 데이터베이스에서 생성된 아카이브 로그의 지속적인 피드를 만들고 복원된 데이터베이스에서 다시 재생하여 두 로그 모두 동일한 상태에 가깝게 유지하는 것이 목표입니다. 전환 시간이 되면 소스 데이터베이스가 완전히 종료되고 최종 아카이브 로그가 복사되고 경우에 따라 재실행 로그가 재생됩니다. 리두 로그에는 커밋된 최종 트랜잭션이 포함될 수 있으므로 리두 로그도 고려해야 합니다.

이러한 로그를 전송하고 재생한 후에는 두 데이터베이스가 서로 일관됩니다. 이 시점에서 대부분의 고객은 몇 가지 기본 테스트를 수행합니다. 마이그레이션 프로세스 중에 오류가 발생하면 로그 재생에서 오류를 보고하고 실패합니다. 알려진 쿼리 또는 응용 프로그램 기반 작업을 기반으로 몇 가지 빠른 테스트를 수행하여 구성이 최적화되었는지 확인하는 것이 좋습니다. 또한 원래 데이터베이스를 종료하기 전에 최종 테스트 테이블을 하나 만들어 마이그레이션된 데이터베이스에 있는지 확인하는 것이 일반적입니다. 이 단계를 수행하면 최종 로그 동기화 중에 오류가 발생하지 않습니다.

단순한 로그 전달 마이그레이션은 원본 데이터베이스와 관련하여 대역 외 방식으로 구성할 수 있으므로 업무상 중요한 데이터베이스에 특히 유용합니다. 소스 데이터베이스에 대한 구성 변경이 필요하지 않으며 마이그레이션 환경의 복원 및 초기 구성은 운영 작업에 영향을 미치지 않습니다. 로그 전달이 구성된 후 운영 서버에 일부 입출력 요구 사항이 배치됩니다. 그러나 로그 전달은 아카이브 로그의 단순 순차 읽기로 구성되므로 운영 데이터베이스 성능에 영향을 미칠 가능성은 낮습니다.

로그 전달은 장거리, 높은 변경률 마이그레이션 프로젝트에 특히 유용한 것으로 입증되었습니다. 한 예로, 하나의 220TB 데이터베이스가 약 500마일 떨어진 새로운 위치로 마이그레이션되었습니다. 변경률이 매우 높았고 보안 제한으로 인해 네트워크 연결이 사용되지 않았습니다. 로그 배송은 테이프 및 택배사를 사용하여 수행되었습니다. 소스 데이터베이스의 복제본은 아래에 설명된 절차를 사용하여 초기에 복원되었습니다. 그런 다음 최종 테이프 세트가 제공되고 로그가 복제 데이터베이스에 적용된 전환 시간까지 택배사에 의해 로그가 매주 배송되었습니다.

Oracle DataGuard

경우에 따라 전체 DataGuard 환경이 보장됩니다. DataGuard라는 용어를 사용하여 로그 전달 또는 대기 데이터베이스 구성을 참조하는 것은 올바르지 않습니다. Oracle DataGuard는 데이터베이스 복제 관리를 위한 포괄적인 프레임워크이지만 복제 기술은 아닙니다. 마이그레이션 과정에서 완전한 DataGuard 환경의 주된 이점은 한 데이터베이스에서 다른 데이터베이스로 투명하게 전환하는 것입니다. 또한 새로운 환경의 성능 또는 네트워크 연결 문제와 같은 문제가 발견될 경우 DataGuard는 원래 데이터베이스로 투명하게 전환할 수 있습니다. 완전히 구성된 DataGuard 환경에서는 애플리케이션이 기본 데이터베이스 위치의 변경을 감지할 수 있도록 데이터베이스 계층뿐만 아니라 응용 프로그램도 구성해야 합니다. 일반적으로 DataGuard를 사용하여 마이그레이션을 완료할 필요는 없지만 일부 고객은 내부에서 광범위한 DataGuard 전문 지식을 보유하고 있으며 마이그레이션 작업에 이미 의존하고 있습니다.

재건축

앞서 설명한 것처럼 스토리지 어레이의 고급 기능을 활용하려면 데이터베이스 레이아웃을 변경해야 하는 경우가 있습니다. 또한 ASM에서 NFS 파일 시스템으로 이동하는 것과 같은 스토리지 프로토콜이 변경될 경우 파일 시스템 레이아웃이 변경될 수도 있습니다.

DataGuard를 비롯한 로그 전달 방법의 주요 이점 중 하나는 복제 대상이 소스와 일치하지 않아도 된다는 것입니다.

로그 전달 방식을 사용하여 ASM에서 일반 파일 시스템으로 마이그레이션하거나 그 반대로 마이그레이션하는 데 문제가 없습니다. 대상 위치에서 데이터 파일의 정확한 레이아웃을 변경하여 플러그형 데이터베이스(PDB) 기술의 사용을 최적화하거나 특정 파일에 대해 선택적으로 QoS 제어를 설정할 수 있습니다. 즉, 로그 전달을 기반으로 하는 마이그레이션 프로세스를 통해 데이터베이스 스토리지 레이아웃을 쉽고 안전하게 최적화할 수 있습니다.

서버 리소스

데이터베이스 수준 마이그레이션의 한 가지 제한 사항은 보조 서버의 필요성입니다. 이 두 번째 서버를 사용하는 방법에는 두 가지가 있습니다.

1. 두 번째 서버를 데이터베이스의 영구적인 새 홈으로 사용할 수 있습니다.
2. 두 번째 서버를 임시 스테이징 서버로 사용할 수 있습니다. 새 스토리지로의 데이터 마이그레이션이 완료되고 테스트된 후 LUN 또는 NFS 파일 시스템이 스테이징 서버에서 분리되어 원래 서버에 다시 연결됩니다.

첫 번째 옵션은 가장 쉽지만 매우 강력한 서버가 필요한 대규모 환경에서는 이 옵션을 사용하는 것이 불가능할 수 있습니다. 두 번째 옵션은 파일 시스템을 원래 위치로 다시 재배치하기 위해 추가 작업이 필요합니다. 이는 파일 시스템을 스테이징 서버에서 마운트 해제하고 원래 서버에 다시 마운트할 수 있기 때문에 NFS를 스토리지 프로토콜로 사용하는 간단한 작업입니다.

블록 기반 파일 시스템은 FC 조닝 또는 iSCSI 이니시에이터를 업데이트하기 위해 추가 작업이 필요합니다. 대부분의 논리적 볼륨 관리자(ASM 포함)에서는 원래 서버에서 LUN을 사용할 수 있게 되면 LUN이 자동으로 감지되어 온라인 상태로 전환됩니다. 그러나 일부 파일 시스템 및 LVM 구현에서는 데이터를 내보내고 가져오는 데 더 많은 작업이 필요할 수 있습니다. 정확한 절차는 다양할 수 있지만 일반적으로 마이그레이션을 완료하고 원래 서버에서 데이터를 다시 저장하는 간단하고 반복 가능한 절차를 설정하는 것이 쉽습니다.

단일 서버 환경 내에서 로그 전달을 설정하고 데이터베이스를 복제할 수 있지만 새 인스턴스에는 로그를 재생하기 위한 다른 프로세스 SID가 있어야 합니다. SID가 다른 프로세스 ID의 다른 집합에서 데이터베이스를 임시로 가져온 후 나중에 변경할 수 있습니다. 하지만 이렇게 하면 복잡한 관리 작업이 많이 발생할 수 있으며 데이터베이스 환경에 사용자 오류가 발생할 위험이 있습니다.

호스트 레벨 마이그레이션

호스트 레벨에서 데이터를 마이그레이션한다는 것은 호스트 운영 체제와 관련 유틸리티를 사용하여 마이그레이션을 완료하는 것을 의미합니다. 이 프로세스에는 Oracle RMAN 및 Oracle ASM을 비롯하여 데이터를 복사하는 모든 유틸리티가 포함됩니다.

데이터 복사

단순 복사 작업의 값은 과소 평가되지 않아야 합니다. 오늘날의 네트워크 인프라는 초당 기가바이트 단위의 속도로 데이터를 이동할 수 있으며 파일 복사 작업은 효율적인 순차적 읽기 및 쓰기 I/O를 기반으로 합니다. 로그 전달과 비교할 때 호스트 복제 작업에서 더 많은 중단이 불가피하지만 마이그레이션은 단순한 데이터 이동 그 이상입니다. 여기에는 일반적으로 네트워킹, 데이터베이스 재시작 시간 및 마이그레이션 후 테스트 변경 사항이 포함됩니다.

데이터를 복사하는 데 필요한 실제 시간은 중요하지 않을 수 있습니다. 또한 원본 데이터를 그대로 유지하므로 복제 작업은 보장된 백아웃 경로를 유지합니다. 마이그레이션 프로세스 중에 문제가 발생하면 원본 데이터가 있는 원본 파일 시스템을 다시 활성화할 수 있습니다.

플랫폼 변경

플랫폼 변경이란 CPU 유형의 변경을 의미합니다. 데이터베이스를 기존 Solaris, AIX 또는 HP-UX 플랫폼에서 x86 Linux로 마이그레이션할 경우 CPU 아키텍처의 변경으로 인해 데이터를 다시 포맷해야 합니다. SPARC, IA64 및 전원 CPU는 빅 엔디안 프로세서라고 하는 반면 x86 및 x86_64 아키텍처는 리틀 엔디안이라고 합니다. 따라서 Oracle 데이터 파일 내의 일부 데이터는 사용 중인 프로세서에 따라 순서가 다르게 지정됩니다.

기존에는 DataPump를 사용하여 플랫폼 간에 데이터를 복제해 왔습니다. 데이터 덤프는 대상 데이터베이스에서 보다 빠르게 가져올 수 있는 특수한 유형의 논리적 데이터 내보내기를 만드는 유틸리티입니다. DataPump 는 데이터의 논리적 복사본을 만들기 때문에 프로세서 엔디언의 종속성을 남깁니다. 데이터덤프는 여전히 일부 고객이 플랫폼 재구축을 위해 사용하고 있지만 Oracle 11g에서는 더욱 빠른 옵션인 교차 플랫폼 전송 테이블스페이스를 사용할 수 있게 되었습니다. 이렇게 하면 테이블스페이스를 다른 엔디언 형식으로 변환할 수 있습니다. 이것은 물리적 바이트를 논리적 데이터로 변환한 다음 다시 물리적 바이트로 변환해야 하는 DataPump 내보내기보다 더 나은 성능을 제공하는 물리적 변환입니다.

DataPump 및 이식 가능한 테이블스페이스에 대한 자세한 내용은 NetApp 설명서를 참조하십시오. 하지만 NetApp는 새로운 CPU 아키텍처를 사용하여 새 스토리지 시스템 로드로 마이그레이션할 때 고객을 지원하는 경험을 바탕으로 몇 가지 권장 사항을 제시합니다.

- DataPump를 사용 중인 경우 마이그레이션을 완료하는 데 필요한 시간을 테스트 환경에서 측정해야 합니다. 고객은 마이그레이션을 완료하는 데 필요한 시간에 놀라기도 합니다. 이와 같이 예기치 않은 추가 다운타임은 운영 중단을 일으킬 수 있습니다.
- 많은 고객들이 교차 플랫폼 전송 가능 테이블스페이스는 데이터 변환이 필요하지 않다고 잘못 생각합니다. 엔디언이 다른 CPU를 사용하는 경우 RMAN이 사용됩니다 convert 데이터 파일에 대한 작업은 미리 수행해야 합니다. 이것은 즉각적인 작업이 아닙니다. 경우에 따라 서로 다른 데이터 파일에서 여러 스레드가 작동하므로 변환 프로세스가 빨라질 수 있지만 변환 프로세스를 피할 수는 없습니다.

논리적 볼륨 관리자 기반 마이그레이션

LVM은 하나 이상의 LUN 그룹을 만들어 일반적으로 익스텐트라고 하는 작은 단위로 분할하는 방식으로 작동합니다. 그런 다음 익스텐트 풀이 기본적으로 가상화된 논리적 볼륨을 생성하기 위한 소스로 사용됩니다. 이 가상화 계층은 다음과 같은 다양한 방식으로 가치를 제공합니다.

- 논리적 볼륨은 여러 LUN에서 그린 익스텐트를 사용할 수 있습니다. 논리적 볼륨에 파일 시스템을 생성할 때 모든 LUN의 전체 성능을 사용할 수 있습니다. 또한 볼륨 그룹에 모든 LUN의 로드가 짝수일 뿐이므로 성능이 더욱 예측 가능합니다.
- 논리적 볼륨의 크기는 익스텐트를 추가하거나 경우에 따라 제거할 수 있습니다. 논리적 볼륨에서 파일 시스템의 크기를 조정하는 작업은 일반적으로 중단되지 않습니다.
- 기본 익스텐트를 이동하여 논리적 볼륨을 운영 중단 없이 마이그레이션할 수 있습니다.

LVM을 사용한 마이그레이션은 익스텐트 이동 또는 익스텐트 미러링/디머러링의 두 가지 방법 중 하나로 작동합니다. LVM 마이그레이션은 효율적인 대규모 블록 순차적 I/O를 사용하며 성능 문제는 거의 발생하지 않습니다. 이 문제가 발생할 경우 일반적으로 I/O 속도를 제한하는 옵션이 있습니다. 이렇게 하면 마이그레이션을 완료하는 데 필요한 시간이 길어지고 호스트 및 스토리지 시스템의 I/O 부담이 줄어듭니다.

미러 및 미러

AIX LVM과 같은 일부 볼륨 관리자는 사용자가 각 익스텐트의 복제본 수를 지정하고 각 복제본을 호스팅하는 디바이스를 제어할 수 있도록 합니다. 마이그레이션은 기존의 논리적 볼륨을 만들고 기본 익스텐트를 새 볼륨에 미러링하고 복사본이 동기화될 때까지 기다린 다음 이전 복사본을 삭제하여 수행됩니다. 백업 경로가 필요한 경우 미러 복사본이 삭제되기 전에 원본 데이터의 스냅샷을 생성할 수 있습니다. 또는 포함된 미러 복제본을 강제로 삭제하기 전에 서버를 잠시 종료하여 원래 LUN을 마스킹할 수 있습니다. 이렇게 하면 복구 가능한 데이터 복사본이 원래 위치에 보존됩니다.

익스텐트 마이그레이션

거의 모든 볼륨 관리자는 익스텐트의 마이그레이션을 허용하며 경우에 따라서는 여러 옵션이 존재하기도 합니다. 예를 들어 일부 볼륨 관리자에서는 관리자가 특정 논리적 볼륨의 개별 익스텐트를 이전 스토리지에서 새 스토리지로 재배치할

수 있습니다. Linux LVM2와 같은 볼륨 관리자는 를 제공합니다 `pvmove` 지정된 LUN 디바이스의 모든 익스텐트를 새 LUN으로 재배치하는 명령입니다. 이전 LUN을 이동한 후 제거할 수 있습니다.



운영 시 가장 큰 위험은 구성에서 사용되지 않은 오래된 LUN을 제거하는 것입니다. FC 조닝을 변경하고 오래된 LUN 디바이스를 제거할 때는 특히 주의해야 합니다.

Oracle 자동 스토리지 관리

Oracle ASM은 논리 볼륨 관리자와 파일 시스템이 결합된 시스템입니다. 상위 수준에서 Oracle ASM은 LUN 모음을 가져와 작은 할당 단위로 분할하고 ASM 디스크 그룹이라고 하는 단일 볼륨으로 제공합니다. ASM에는 이중화 수준을 설정하여 디스크 그룹을 미러링하는 기능도 포함되어 있습니다. 볼륨은 미러링되지 않은(외부 중복), 미러링(일반 중복) 또는 3웨이 미러링(높은 중복)일 수 있습니다. 이중화 수준은 생성 후 변경할 수 없기 때문에 설정 시 주의해야 한다.

ASM은 파일 시스템 기능도 제공합니다. 파일 시스템이 호스트에서 직접 표시되지 않지만 Oracle 데이터베이스는 ASM 디스크 그룹에서 파일과 디렉토리를 생성, 이동 및 삭제할 수 있습니다. 또한 `asmcmd` 유틸리티를 사용하여 구조를 탐색할 수도 있습니다.

다른 LVM 구현과 마찬가지로 Oracle ASM은 사용 가능한 모든 LUN에서 각 파일의 I/O를 스트라이핑 및 로드 밸런싱을 통해 I/O 성능을 최적화합니다. 둘째, 기본 익스텐트를 재배치하여 ASM 디스크 그룹의 크기 조정과 마이그레이션을 모두 수행할 수 있습니다. Oracle ASM은 재조정 작업을 통해 프로세스를 자동화합니다. 새로운 LUN이 ASM 디스크 그룹에 추가되고 기존 LUN이 삭제되어 익스텐트 재배치와 디스크 그룹에서 제거된 LUN의 후속 드롭이 트리거됩니다. 이 프로세스는 가장 검증된 마이그레이션 방법 중 하나이며, 투명한 마이그레이션을 제공하는 ASM의 신뢰성이 가장 중요한 기능일 수 있습니다.



Oracle ASM의 미러링 수준은 고정되어 있으므로 미러 및 미러 마이그레이션 방법과 함께 사용할 수 없습니다.

스토리지 레벨 마이그레이션

스토리지 수준 마이그레이션은 애플리케이션 및 운영 체제 수준 모두에서 마이그레이션을 수행하는 것을 의미합니다. 과거에는 네트워크 수준에서 LUN을 복제할 특수 장치를 사용하기도 했지만 이제는 ONTAP에서 기본적으로 제공하는 이러한 기능을 사용할 수 있습니다.

SnapMirror를 참조하십시오

NetApp 시스템 간 데이터베이스 마이그레이션은 NetApp SnapMirror 데이터 복제 소프트웨어를 통해 거의 보편적으로 수행됩니다. 이 프로세스에는 마이그레이션할 볼륨의 미러 관계를 설정하고 볼륨이 동기화될 수 있도록 한 다음 컷오버 기간을 기다리는 작업이 포함됩니다. 소스 데이터베이스가 도착하면 소스 데이터베이스가 종료되고 최종 미러 업데이트가 한 번 수행되며 미러가 중단됩니다. 그러면 포함된 NFS 파일 시스템 디렉토리를 마운트하거나 포함된 LUN을 검색하고 데이터베이스를 시작하여 복제본 볼륨을 사용할 수 있습니다.

단일 ONTAP 클러스터 내에서 볼륨을 재배치하는 것은 마이그레이션으로 간주되는 것이 아니라 일상적인 마이그레이션으로 간주됩니다 `volume move` 작동. SnapMirror는 클러스터 내의 데이터 복제 엔진으로 사용됩니다. 이 프로세스는 완전히 자동화되어 있습니다. LUN 매핑이나 NFS 익스포트 권한과 같은 볼륨 특성을 볼륨 자체와 함께 이동할 때 수행해야 할 추가 마이그레이션 단계는 없습니다. 재할당은 호스트 작업의 중단 없이 수행됩니다. 경우에 따라 새로 재배치된 데이터에 가장 효율적인 방식으로 액세스할 수 있도록 네트워크 액세스를 업데이트해야 하지만, 이러한 작업은 중단되지 않습니다.

FLI(Foreign LUN Import)

FLI는 8.3 이상을 실행하는 Data ONTAP 시스템에서 다른 스토리지 어레이의 기존 LUN을 마이그레이션할 수 있는 기능입니다. 절차는 간단합니다. ONTAP 시스템은 다른 SAN 호스트처럼 기존 스토리지 시스템에 조닝됩니다. 그런

다음 Data ONTAP는 원하는 레거시 LUN을 제어하고 기본 데이터를 마이그레이션합니다. 또한 가져오기 프로세스에서는 데이터가 마이그레이션될 때 새 볼륨의 효율성 설정을 사용합니다. 즉, 마이그레이션 프로세스 중에 데이터를 인라인으로 압축 및 중복제거할 수 있습니다.

Data ONTAP 8.3에서 FLI를 처음 구현하면 오프라인 마이그레이션만 허용되었습니다. 이는 매우 빠른 전송이었지만 마이그레이션이 완료될 때까지 LUN 데이터를 사용할 수 없다는 것을 의미합니다. 온라인 마이그레이션은 Data ONTAP 8.3.1에서 도입되었습니다. 이러한 종류의 마이그레이션은 전송 프로세스 중에 ONTAP에서 LUN 데이터를 제공할 수 있으므로 작업 중단이 최소화됩니다. ONTAP를 통해 LUN을 사용하도록 호스트를 다시 조닝하는 동안 중단이 짧게 발생합니다. 그러나 이러한 변경이 이루어지면 데이터에 다시 액세스할 수 있고 마이그레이션 프로세스 내내 계속 액세스할 수 있습니다.

읽기 입출력은 복제 작업이 완료될 때까지 ONTAP를 통해 프록시되고 쓰기 입출력은 외부 및 ONTAP LUN 모두에 동기식으로 기록됩니다. 관리자가 전체 컷오버를 실행하여 외부 LUN을 해제하고 더 이상 쓰기를 복제하지 않는 한 두 LUN 복사본이 이 방식으로 동기화된 상태로 유지됩니다.

FLI는 FC와 함께 사용하도록 설계되었지만 iSCSI로 변경하려는 경우 마이그레이션이 완료된 후 마이그레이션된 LUN을 iSCSI LUN으로 쉽게 다시 매핑할 수 있습니다.

FLI의 기능 중 하나는 자동 정렬 감지 및 조정입니다. 여기서 정렬이란 LUN 장치의 파티션을 의미합니다. 최적의 성능을 얻으려면 I/O를 4K 블록에 맞춰 정렬해야 합니다. 파티션이 4K의 배수가 아닌 오프셋에 배치되면 성능이 저하됩니다.

정렬의 두 번째 측면은 파티션 오프셋을 조정하여 수정할 수 없는 파일 시스템 블록 크기입니다. 예를 들어, ZFS 파일 시스템의 기본 내부 블록 크기는 512바이트입니다. AIX를 사용하는 다른 고객은 512 또는 1,024바이트 블록 크기의 JFS2 파일 시스템을 생성하는 경우가 있습니다. 파일 시스템이 4K 경계에 맞춰 정렬될 수 있지만 해당 파일 시스템 내에서 생성된 파일은 그렇지 않고 성능이 저하됩니다.

FLI는 이러한 상황에서 사용해서는 안 됩니다. 마이그레이션 후에 데이터에 액세스할 수 있지만 이로 인해 파일 시스템의 성능이 심각하게 제한됩니다. 일반적으로 ONTAP에서 랜덤 덮어쓰기 워크로드를 지원하는 모든 파일 시스템은 4K 블록 크기를 사용해야 합니다. 이 워크로드는 데이터베이스 데이터 파일 및 VDI 구축과 같은 워크로드에 주로 적용됩니다. 블록 크기는 관련 호스트 운영 체제 명령을 사용하여 확인할 수 있습니다.

예를 들어, AIX에서는 블록 크기를 로 볼 수 있습니다 `lsfs -q`. Linux를 사용하면 `xfs_info` 및 `tune2fs` 에 사용할 수 있습니다 `xfs` 및 `ext3/ext4` 있습니다. 와 함께 ``zfs`` 명령은 `입니다 `zdb -C`.

블록 크기를 제어하는 매개 변수는 `ashift` 입니다 일반적으로 기본값은 9이며, 이는 2의 9 또는 512바이트를 의미합니다. 최적의 성능을 위해 `ashift` 값은 `12(2-12=4K)`여야 합니다. 이 값은 `zpool`이 생성될 때 설정되며 변경할 수 없습니다. 즉, 가 포함된 데이터 `zpool`이 됩니다 `ashift 12`가 아닌 경우 데이터를 새로 생성된 `zpool`으로 마이그레이션해야 합니다.

Oracle ASM은 기본 블록 크기를 가지고 있지 않습니다. 유일한 요구 사항은 ASM 디스크가 구축된 파티션이 올바르게 정렬되어야 한다는 것입니다.

7-Mode 전환 툴

7MTT(7-Mode 전환 툴)는 대규모 7-Mode 구성을 ONTAP로 마이그레이션하는 데 사용되는 자동화 유틸리티입니다. 대부분의 데이터베이스 고객은 전체 스토리지 공간을 재배치하지 않고 데이터베이스를 기준으로 환경을 마이그레이션하므로 다른 방법을 더욱 쉽게 찾을 수 있습니다. 또한 데이터베이스는 대규모 스토리지 환경에 포함되는 경우가 많습니다. 따라서 데이터베이스는 종종 개별적으로 마이그레이션되며, 7MTT를 사용하여 나머지 환경을 이동할 수 있습니다.

복잡한 데이터베이스 환경을 위한 스토리지 시스템을 보유한 고객 수는 소규모지만 상당수가 있습니다. 이러한 환경에는 많은 볼륨, 스냅샷 및 내보내기 권한, LUN 이니시에이터 그룹, 사용자 권한 및 Lightweight Directory Access Protocol 구성과 같은 수많은 구성 세부 정보가 포함될 수 있습니다. 이런 경우에는 7MTT의 자동화 기능을 사용하여 마이그레이션을 단순화할 수 있습니다.

7MTT는 다음 2가지 모드 중 하나로 작동할 수 있습니다.

- * CBT(Copy-Based Transition). * CBT를 사용하는 7MTT는 새로운 환경의 기존 7-Mode 시스템에서 SnapMirror 볼륨을 설정합니다. 데이터가 동기화되면 7MTT가 컷오버 프로세스를 오케스트레이션합니다.
- * CFT(Copy-Free Transition) * CFT를 지원하는 7MTT는 기존 7-Mode 디스크 쉘프의 데이터 이동 없이 변환을 기반으로 합니다. 데이터는 복사되지 않으며 기존 디스크 쉘프를 재사용할 수 있습니다. 기존 데이터 보호 및 스토리지 효율성 구성이 그대로 유지됩니다.

이 두 옵션 간의 주된 차이점은 복사가 필요 없는 전환은 원래의 7-Mode HA 쌍에 연결된 모든 디스크 쉘프를 새로운 환경으로 재배포해야 하는 큰 방식이라는 것입니다. 쉘프의 하위 집합을 이동할 수 있는 옵션은 없습니다. 복사 기반 접근 방식에서는 선택한 볼륨을 이동할 수 있습니다. 또한 디스크 쉘프를 재구성하고 메타데이터를 변환하는 데 연결된 연결이 필요하므로 무복사 전환으로 컷오버 기간도 길어질 수 있습니다. 현장 경험에 비추어 볼 때, NetApp은 디스크 쉘프를 재배포하고 재설정하는 데 1시간, 메타데이터 변환에 15분에서 2시간 동안 사용할 것을 권장합니다.

Oracle 데이터 파일 마이그레이션

단일 명령으로 개별 Oracle 데이터 파일을 이동할 수 있습니다.

예를 들어, 다음 명령은 파일 시스템에서 데이터 파일 IOPST.dbf를 이동합니다 /oradata2 파일 시스템으로 /oradata3.

```
SQL> alter database move datafile '/oradata2/NTAP/IOPS002.dbf' to
'/oradata3/NTAP/IOPS002.dbf';
Database altered.
```

이 방법으로 데이터 파일을 이동하는 것은 느낄 수 있지만 일반적으로 일상적인 데이터베이스 워크로드에 지장을 줄 만큼 I/O가 충분히 생성하지 않아야 합니다. 반면 ASM 재조정을 통한 마이그레이션은 훨씬 빠르게 실행할 수 있지만 데이터가 이동되는 동안 전체 데이터베이스의 속도가 느려질 수 있습니다.

데이터 파일을 이동한 시간은 테스트 데이터 파일을 만든 다음 이동하는 방법으로 손쉽게 측정할 수 있습니다. 작업에 대해 경과된 시간은 v\$ 세션 데이터에 기록됩니다.

```
SQL> set linesize 300;
SQL> select elapsed_seconds||':'||message from v$session_longops;
ELAPSED_SECONDS||':'||MESSAGE
-----
-----
351:Online data file move: data file 8: 22548578304 out of 22548578304
bytes done
SQL> select bytes / 1024 / 1024 /1024 as GB from dba_data_files where
FILE_ID = 8;
          GB
-----
          21
```

이 예에서는 데이터 파일 8로 이동했습니다. 데이터 파일 8은 21GB이며 마이그레이션하는 데 6분 정도 걸렸습니다. 필요한 시간은 스토리지 시스템의 기능, 스토리지 네트워크 및 마이그레이션 시 발생하는 전체 데이터베이스 활동에

따라 달라집니다.

로그 전달을 통한 Oracle 데이터베이스 마이그레이션

로그 전달을 사용하는 마이그레이션의 목표는 새 위치에 원본 데이터 파일의 복사본을 만든 다음 새 환경에 변경 사항을 전달하는 방법을 설정하는 것입니다.

설정된 후에는 로그 전송 및 재생을 자동화하여 복제 데이터베이스를 소스와 대부분 동기화된 상태로 유지할 수 있습니다. 예를 들어, Cron 작업은 (a) 가장 최근의 로그를 새 위치로 복사하고 (b) 15분마다 재생하도록 예약할 수 있습니다. 이렇게 하면 아카이브 로그를 15분 이상 재생해야 하므로 전환 시 작업 중단이 최소화됩니다.

아래에 나와 있는 절차는 기본적으로 데이터베이스 클론 작업입니다. 표시된 로직은 NetApp SnapManager for Oracle(SMO) 및 NetApp SnapCenter Oracle 플러그인 내의 엔진과 유사합니다. 일부 고객은 맞춤형 클론 복제 작업을 위해 스크립트 또는 WFA 워크플로우에 표시된 절차를 사용했습니다. 이 절차는 SMO 또는 SnapCenter를 사용하는 것보다 수동적이지만, ONTAP의 데이터 관리 API로 인해 프로세스가 더욱 간소화됩니다.

로그 전달 - 파일 시스템을 파일 시스템으로 전송합니다

이 예제에서는 Waffle이라는 데이터베이스를 일반 파일 시스템에서 다른 서버에 있는 다른 일반 파일 시스템으로 마이그레이션하는 방법을 보여 줍니다. 또한 SnapMirror를 사용하여 데이터 파일의 신속한 복사본을 만드는 방법을 보여 주지만 이것이 전체 절차의 필수 요소가 아닙니다.

데이터베이스 백업을 만듭니다

첫 번째 단계는 데이터베이스 백업을 만드는 것입니다. 특히 이 절차를 수행하려면 아카이브 로그 재생에 사용할 수 있는 데이터 파일 세트가 필요합니다.

방법입니다

이 예에서 소스 데이터베이스는 ONTAP 시스템에 있습니다. 데이터베이스 백업을 만드는 가장 간단한 방법은 스냅샷을 사용하는 것입니다. 데이터베이스는 가 있는 동안 몇 초 동안 핫 백업 모드로 전환됩니다 snapshot create 이 작업은 데이터 파일을 호스팅하는 볼륨에서 실행됩니다.

```
SQL> alter database begin backup;
Database altered.
```

```
Cluster01::*> snapshot create -vserver vserver1 -volume jfsc1_oradata
hotbackup
Cluster01::*>
```

```
SQL> alter database end backup;
Database altered.
```

그 결과 라는 디스크에 스냅샷이 생성됩니다 hotbackup 핫 백업 모드에 있는 동안 데이터 파일의 이미지가 포함됩니다. 적절한 아카이브 로그와 결합하여 데이터 파일의 일관성을 유지할 경우 이 스냅샷의 데이터를 복구 또는 클론의 기준으로 사용할 수 있습니다. 이 경우 새 서버에 복제됩니다.

새 환경으로 복원합니다

이제 새 환경에서 백업을 복원해야 합니다. 이는 Oracle RMAN, NetBackup 같은 백업 애플리케이션에서 복원 또는 핫 백업 모드에 있었던 데이터 파일의 간단한 복사 작업을 비롯하여 다양한 방법으로 수행할 수 있습니다.

이 예에서는 SnapMirror를 사용하여 스냅샷 핫 백업을 새 위치에 복제합니다.

1. 스냅샷 데이터를 수신할 새 볼륨을 생성합니다. 에서 미러링을 초기화합니다 jfsc1_oradata 를 선택합니다 vol_oradata.

```
Cluster01::*> volume create -vserver vserver1 -volume vol_oradata
-aggregate data_01 -size 20g -state online -type DP -snapshot-policy
none -policy jfsc3
[Job 833] Job succeeded: Successful
```

```
Cluster01::*> snapmirror initialize -source-path vserver1:jfsc1_oradata
-destination-path vserver1:vol_oradata
Operation is queued: snapmirror initialize of destination
"vserver1:vol_oradata".
Cluster01::*> volume mount -vserver vserver1 -volume vol_oradata
-junction-path /vol_oradata
Cluster01::*>
```

2. 동기화가 완료되었음을 나타내는 SnapMirror가 상태를 설정한 후, 특히 원하는 스냅샷을 기반으로 미러를 업데이트합니다.

```
Cluster01::*> snapmirror show -destination-path vserver1:vol_oradata
-fields state
source-path          destination-path      state
-----
vserver1:jfsc1_oradata vserver1:vol_oradata SnapMirrored
```

```
Cluster01::*> snapmirror update -destination-path vserver1:vol_oradata
-source-snapshot hotbackup
Operation is queued: snapmirror update of destination
"vserver1:vol_oradata".
```

3. 동기화를 확인하려면 을 참조하십시오 newest-snapshot 미러 볼륨의 필드입니다.

```
Cluster01::*> snapmirror show -destination-path vserver1:vol_oradata
-fields newest-snapshot
source-path          destination-path      newest-snapshot
-----
vserver1:jfsc1_oradata vserver1:vol_oradata hotbackup
```

4. 그런 다음 미러가 파손될 수 있습니다.

```
Cluster01::> snapmirror break -destination-path vserver1:vol_oradata
Operation succeeded: snapmirror break for destination
"vserver1:vol_oradata".
Cluster01::>
```

5. 새 파일 시스템을 마운트합니다. 블록 기반 파일 시스템을 사용하면 사용 중인 LVM에 따라 정확한 절차가 달라집니다. FC 조닝 또는 iSCSI 연결을 구성해야 합니다. LUN에 대한 연결이 설정된 후 Linux와 같은 명령이 표시됩니다 pvscan ASM이 검색할 수 있도록 올바르게 구성해야 하는 볼륨 그룹 또는 LUN을 찾는 데 필요한 수도 있습니다.

이 예에서는 단순 NFS 파일 시스템이 사용됩니다. 이 파일 시스템은 직접 마운트할 수 있습니다.

```
fas8060-nfs1:/vol_oradata          19922944    1639360    18283584    9%
/oradata
fas8060-nfs1:/vol_logs            9961472     128        9961344     1%
/logs
```

컨트롤 파일 만들기 템플릿을 만듭니다

그런 다음 컨트롤 파일 템플릿을 만들어야 합니다. 를 클릭합니다 backup controlfile to trace 명령은 텍스트 명령을 만들어 컨트롤 파일을 다시 만듭니다. 이 기능은 경우에 따라 백업에서 데이터베이스를 복원하는 데 유용할 수 있으며 데이터베이스 클론 생성과 같은 작업을 수행하는 스크립트와 함께 사용됩니다.

1. 다음 명령의 출력은 마이그레이션된 데이터베이스에 대한 컨트롤 파일을 다시 생성하는 데 사용됩니다.

```
SQL> alter database backup controlfile to trace as '/tmp/waffle.ctrl';
Database altered.
```

2. 컨트롤 파일을 만든 후 새 서버에 파일을 복사합니다.

```
[oracle@jfsc3 tmp]$ scp oracle@jfsc1:/tmp/waffle.ctrl /tmp/
oracle@jfsc1's password:
waffle.ctrl                                100% 5199
5.1KB/s  00:00
```

매개 변수 파일을 백업합니다

매개 변수 파일도 새 환경에 필요합니다. 가장 간단한 방법은 현재 spfile 또는 pfile 에서 pfile 을 만드는 것입니다. 이 예에서는 소스 데이터베이스가 spfile을 사용하고 있습니다.

```
SQL> create pfile='/tmp/waffle.tmp.pfile' from spfile;
File created.
```

ORATAB 항목을 만듭니다

오라타브 항목의 생성은 오라타브 같은 유틸리티의 적절한 기능을 위해 필요합니다. ORATAB 항목을 만들려면 다음 단계를 완료합니다.

```
WAFFLE:/orabin/product/12.1.0/dbhome_1:N
```

디렉토리 구조를 준비합니다

필요한 디렉터리가 없는 경우 해당 디렉터리를 만들어야 합니다. 그렇지 않으면 데이터베이스 시작 절차가 실패합니다. 디렉터리 구조를 준비하려면 다음과 같은 최소 요구 사항을 완료하십시오.

```
[oracle@jpsc3 ~]$ . oraenv
ORACLE_SID = [oracle] ? WAFFLE
The Oracle base has been set to /orabin
[oracle@jpsc3 ~]$ cd $ORACLE_BASE
[oracle@jpsc3 orabin]$ cd admin
[oracle@jpsc3 admin]$ mkdir WAFFLE
[oracle@jpsc3 admin]$ cd WAFFLE
[oracle@jpsc3 WAFFLE]$ mkdir adump dpdump pfile scripts xdb_wallet
```

매개 변수 파일 업데이트

1. 매개 변수 파일을 새 서버에 복사하려면 다음 명령을 실행합니다. 기본 위치는 입니다 \$ORACLE_HOME/dbs 디렉토리. 이 경우 pfile은 어디에나 배치할 수 있습니다. 마이그레이션 프로세스의 중간 단계로만 사용되고 있습니다.

```
[oracle@jpsc3 admin]$ scp oracle@jpsc1:/tmp/waffle.tmp.pfile
$ORACLE_HOME/dbs/waffle.tmp.pfile
oracle@jpsc1's password:
waffle.pfile                                100%  916
0.9KB/s   00:00
```

1. 필요에 따라 파일을 편집합니다. 예를 들어 아카이브 로그 위치가 변경된 경우 새 위치를 반영하도록 pfile을 변경해야 합니다. 이 예제에서는 제어 파일만 재배치되고 일부는 로그 및 데이터 파일 시스템 간에 배포됩니다.

```
[root@jfscl tmp]# cat waffle.pfile
WAFFLE.__data_transfer_cache_size=0
WAFFLE.__db_cache_size=507510784
WAFFLE.__java_pool_size=4194304
WAFFLE.__large_pool_size=20971520
WAFFLE.__oracle_base='/orabin'#ORACLE_BASE set from environment
WAFFLE.__pga_aggregate_target=268435456
WAFFLE.__sga_target=805306368
WAFFLE.__shared_io_pool_size=29360128
WAFFLE.__shared_pool_size=234881024
WAFFLE.__streams_pool_size=0
*.audit_file_dest='/orabin/admin/WAFFLE/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='/oradata//WAFFLE/control01.ctl','/oradata//WAFFLE/control02.ctl'
*.control_files='/oradata/WAFFLE/control01.ctl','/logs/WAFFLE/control02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='WAFFLE'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=WAFFLEXDB)'
*.log_archive_dest_1='LOCATION=/logs/WAFFLE/arch'
*.log_archive_format='%t_%s_%r.dbf'
*.open_cursors=300
*.pga_aggregate_target=256m
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=768m
*.undo_tablespace='UNDOTBS1'
```

2. 편집이 완료되면 이 pfile을 기반으로 spfile을 만듭니다.

```
SQL> create spfile from pfile='waffle.tmp.pfile';
File created.
```

컨트롤 파일을 다시 만듭니다

이전 단계에서 의 출력입니다 backup controlfile to trace 새 서버로 복사되었습니다. 필요한 출력의 특정 부분은 입니다 controlfile recreation 명령. 이 정보는 표시된 섹션 아래의 파일에서 찾을 수 있습니다 Set #1. NORESETLOGS. 라인부터 시작합니다 create controlfile reuse database 및 은 단어를 포함해야 합니다 noresetlogs. 세미콜론(;) 문자로 끝납니다.

1. 이 예제 절차에서 파일은 다음과 같이 읽힙니다.

```
CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS  ARCHIVELOG
  MAXLOGFILES 16
  MAXLOGMEMBERS 3
  MAXDATAFILES 100
  MAXINSTANCES 8
  MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/logs/WAFFLE/redo/redo01.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 2 '/logs/WAFFLE/redo/redo02.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 3 '/logs/WAFFLE/redo/redo03.log'  SIZE 50M BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  '/oradata/WAFFLE/system01.dbf',
  '/oradata/WAFFLE/sysaux01.dbf',
  '/oradata/WAFFLE/undotbs01.dbf',
  '/oradata/WAFFLE/users01.dbf'
CHARACTER SET WE8MSWIN1252
;
```

2. 다양한 파일의 새 위치를 반영하기 위해 이 스크립트를 편집합니다. 예를 들어, 높은 I/O를 지원하는 것으로 알려진 특정 데이터 파일은 고성능 스토리지 계층의 파일 시스템으로 리디렉션될 수 있습니다. 다른 경우에는 지정된 PDB의 데이터 파일을 전용 볼륨에 격리하는 것과 같은 관리자의 이유만으로 변경 내용이 변경될 수 있습니다.
3. 이 예에서 는 입니다 DATAFILE 스탠자는 변경되지 않은 상태로 유지되지만 다시 실행 로그는 의 새 위치로 이동됩니다 /redo 아카이브 로그와 공간을 공유하는 대신 /logs.

```
CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS ARCHIVELOG
  MAXLOGFILES 16
  MAXLOGMEMBERS 3
  MAXDATAFILES 100
  MAXINSTANCES 8
  MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/redo/redo01.log' SIZE 50M BLOCKSIZE 512,
  GROUP 2 '/redo/redo02.log' SIZE 50M BLOCKSIZE 512,
  GROUP 3 '/redo/redo03.log' SIZE 50M BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  '/oradata/WAFFLE/system01.dbf',
  '/oradata/WAFFLE/sysaux01.dbf',
  '/oradata/WAFFLE/undotbs01.dbf',
  '/oradata/WAFFLE/users01.dbf'
CHARACTER SET WE8MSWIN1252
;
```

```

SQL> startup nomount;
ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size              331353200 bytes
Database Buffers          465567744 bytes
Redo Buffers                5455872 bytes
SQL> CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS  ARCHIVELOG
 2     MAXLOGFILES 16
 3     MAXLOGMEMBERS 3
 4     MAXDATAFILES 100
 5     MAXINSTANCES 8
 6     MAXLOGHISTORY 292
 7 LOGFILE
 8   GROUP 1 '/redo/redo01.log'  SIZE 50M BLOCKSIZE 512,
 9   GROUP 2 '/redo/redo02.log'  SIZE 50M BLOCKSIZE 512,
10   GROUP 3 '/redo/redo03.log'  SIZE 50M BLOCKSIZE 512
11  -- STANDBY LOGFILE
12  DATAFILE
13    '/oradata/WAFFLE/system01.dbf',
14    '/oradata/WAFFLE/sysaux01.dbf',
15    '/oradata/WAFFLE/undotbs01.dbf',
16    '/oradata/WAFFLE/users01.dbf'
17  CHARACTER SET WE8MSWIN1252
18  ;
Control file created.
SQL>

```

파일이 잘못 배치되거나 매개 변수가 잘못 구성된 경우 수정해야 할 항목을 나타내는 오류가 생성됩니다. 데이터베이스가 마운트되었지만 아직 열려 있지 않으며 사용 중인 데이터 파일이 핫 백업 모드로 표시되어 있기 때문에 열 수 없습니다. 데이터베이스의 일관성을 유지하기 위해서는 먼저 아카이브 로그를 적용해야 합니다.

초기 로그 복제

데이터 파일의 일관성을 유지하려면 하나 이상의 로그 응답 작업이 필요합니다. 로그를 재생하는 데 사용할 수 있는 옵션은 다양합니다. 경우에 따라 원래 서버의 원래 아카이브 로그 위치를 NFS를 통해 공유할 수 있으며 로그 회신을 직접 수행할 수 있습니다. 다른 경우에는 아카이브 로그를 복사해야 합니다.

예를 들어, 단순 입니다 scp 작업은 소스 서버에서 마이그레이션 서버로 모든 현재 로그를 복사할 수 있습니다.

```

[oracle@jfsc3 arch]$ scp jfsc1:/logs/WAFFLE/arch/* ./
oracle@jfsc1's password:
1_22_912662036.dbf                100%   47MB
47.0MB/s   00:01
1_23_912662036.dbf                100%   40MB
40.4MB/s   00:00
1_24_912662036.dbf                100%   45MB
45.4MB/s   00:00
1_25_912662036.dbf                100%   41MB
40.9MB/s   00:01
1_26_912662036.dbf                100%   39MB
39.4MB/s   00:00
1_27_912662036.dbf                100%   39MB
38.7MB/s   00:00
1_28_912662036.dbf                100%   40MB
40.1MB/s   00:01
1_29_912662036.dbf                100%   17MB
16.9MB/s   00:00
1_30_912662036.dbf                100%   636KB
636.0KB/s   00:00

```

초기 로그 재생

파일이 아카이브 로그 위치에 있으면 명령을 실행하여 재생할 수 있습니다 `recover database until cancel` 그 다음에 응답이 옵니다 `AUTO` 사용 가능한 모든 로그를 자동으로 재생합니다.

```

SQL> recover database until cancel;
ORA-00279: change 382713 generated at 05/24/2016 09:00:54 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_23_912662036.dbf
ORA-00280: change 382713 for thread 1 is in sequence #23
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00279: change 405712 generated at 05/24/2016 15:01:05 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_24_912662036.dbf
ORA-00280: change 405712 for thread 1 is in sequence #24
ORA-00278: log file '/logs/WAFFLE/arch/1_23_912662036.dbf' no longer
needed for
this recovery
...
ORA-00279: change 713874 generated at 05/26/2016 04:26:43 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_31_912662036.dbf
ORA-00280: change 713874 for thread 1 is in sequence #31
ORA-00278: log file '/logs/WAFFLE/arch/1_30_912662036.dbf' no longer
needed for
this recovery
ORA-00308: cannot open archived log '/logs/WAFFLE/arch/1_31_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3

```

최종 아카이브 로그 응답에서 오류를 보고하지만 이는 정상입니다. 로그는 이를 나타냅니다 sqlplus 특정 로그 파일을 찾지만 찾지 못했습니다. 로그 파일이 아직 존재하지 않기 때문일 가능성이 높습니다.

아카이브 로그를 복사하기 전에 소스 데이터베이스를 종료할 수 있는 경우 이 단계는 한 번만 수행해야 합니다. 아카이브 로그가 복사되고 재생된 다음 프로세스를 계속 진행하여 중요한 재실행 로그를 복제하는 컷오버 프로세스로 이동할 수 있습니다.

중분 로그 복제 및 재생

대부분의 경우 마이그레이션은 즉시 수행되지 않습니다. 마이그레이션 프로세스가 완료되기까지 며칠이나 몇 주가 걸릴 수 있습니다. 즉, 로그가 계속해서 복제본 데이터베이스로 전송되고 재생되어야 합니다. 따라서 컷오버가 도착하면 최소한의 데이터를 전송하고 재생해야 합니다.

이러한 작업은 여러 가지 방법으로 스크립팅할 수 있지만 일반적인 방법 중 하나는 일반적인 파일 복제 유틸리티인 rsync를 사용하는 것입니다. 이 유틸리티를 사용하는 가장 안전한 방법은 데몬으로 구성하는 것입니다. 예를 들면, rsyncd.conf 다음 파일은 라는 리소스를 만드는 방법을 보여 줍니다 waffle.arch Oracle 사용자 자격 증명으로 액세스되고 매핑됩니다 /logs/WAFFLE/arch. 가장 중요한 것은 리소스를 읽기 전용으로 설정하여 운영 데이터를 읽을 수는 있지만 변경할 수는 없다는 것입니다.

```
[root@jfscl arch]# cat /etc/rsyncd.conf
[waffle.arch]
  uid=oracle
  gid=dba
  path=/logs/WAFFLE/arch
  read only = true
[root@jfscl arch]# rsync --daemon
```

다음 명령은 새 서버의 아카이브 로그 대상을 rsync 리소스와 동기화합니다 waffle.arch 원래 서버에 있습니다. 를 클릭합니다 t 의 인수입니다 rsync - potg 타임스탬프를 기준으로 파일 목록을 비교하고 새 파일만 복사하도록 합니다. 이 프로세스는 새 서버의 증분 업데이트를 제공합니다. 이 명령은 정기적으로 실행되도록 cron으로 예약할 수도 있습니다.

```

[oracle@jfsc3 arch]$ rsync -potg --stats --progress jfsc1::waffle.arch/*
/logs/WAFFLE/arch/
1_31_912662036.dbf
    650240 100% 124.02MB/s    0:00:00 (xfer#1, to-check=8/18)
1_32_912662036.dbf
    4873728 100% 110.67MB/s    0:00:00 (xfer#2, to-check=7/18)
1_33_912662036.dbf
    4088832 100%  50.64MB/s    0:00:00 (xfer#3, to-check=6/18)
1_34_912662036.dbf
    8196096 100%  54.66MB/s    0:00:00 (xfer#4, to-check=5/18)
1_35_912662036.dbf
    19376128 100%  57.75MB/s    0:00:00 (xfer#5, to-check=4/18)
1_36_912662036.dbf
     71680 100% 201.15kB/s    0:00:00 (xfer#6, to-check=3/18)
1_37_912662036.dbf
    1144320 100%   3.06MB/s    0:00:00 (xfer#7, to-check=2/18)
1_38_912662036.dbf
    35757568 100%  63.74MB/s    0:00:00 (xfer#8, to-check=1/18)
1_39_912662036.dbf
    984576 100%   1.63MB/s    0:00:00 (xfer#9, to-check=0/18)
Number of files: 18
Number of files transferred: 9
Total file size: 399653376 bytes
Total transferred file size: 75143168 bytes
Literal data: 75143168 bytes
Matched data: 0 bytes
File list size: 474
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 204
Total bytes received: 75153219
sent 204 bytes  received 75153219 bytes  150306846.00 bytes/sec
total size is 399653376  speedup is 5.32

```

로그를 수신한 후 재생해야 합니다. 이전 예에서는 sqlplus를 사용하여 수동으로 실행하는 방법을 보여 줍니다 `recover database until cancel` 쉽게 자동화할 수 있는 프로세스입니다. 여기에 표시된 예에서는 에 설명된 스크립트를 사용합니다 "데이터베이스에서 로그를 재생합니다". 스크립트에는 재생 작업이 필요한 데이터베이스를 지정하는 인수를 사용할 수 있습니다. 이렇게 하면 다중 데이터베이스 마이그레이션 작업에 동일한 스크립트를 사용할 수 있습니다.

```

[oracle@jfsc3 logs]$ ./replay.logs.pl WAFFLE
ORACLE_SID = [WAFFLE] ? The Oracle base remains unchanged with value
/orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu May 26 10:47:16 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 713874 generated at 05/26/2016 04:26:43 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_31_912662036.dbf
ORA-00280: change 713874 for thread 1 is in sequence #31
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 814256 generated at 05/26/2016 04:52:30 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_32_912662036.dbf
ORA-00280: change 814256 for thread 1 is in sequence #32
ORA-00278: log file '/logs/WAFFLE/arch/1_31_912662036.dbf' no longer
needed for
this recovery
ORA-00279: change 814780 generated at 05/26/2016 04:53:04 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_33_912662036.dbf
ORA-00280: change 814780 for thread 1 is in sequence #33
ORA-00278: log file '/logs/WAFFLE/arch/1_32_912662036.dbf' no longer
needed for
this recovery
...
ORA-00279: change 1120099 generated at 05/26/2016 09:59:21 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_40_912662036.dbf
ORA-00280: change 1120099 for thread 1 is in sequence #40
ORA-00278: log file '/logs/WAFFLE/arch/1_39_912662036.dbf' no longer
needed for
this recovery
ORA-00308: cannot open archived log '/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options

```

컷오버

새 환경으로 전환할 준비가 되면 아카이브 로그와 redo 로그를 모두 포함하는 하나의 최종 동기화를 수행해야 합니다. 원래 redo 로그 위치를 아직 모르는 경우 다음과 같이 식별할 수 있습니다.

```
SQL> select member from v$logfile;
MEMBER
-----
-----
/logs/WAFFLE/redo/redo01.log
/logs/WAFFLE/redo/redo02.log
/logs/WAFFLE/redo/redo03.log
```

1. 원본 데이터베이스를 종료합니다.
2. 원하는 방법으로 새 서버에서 아카이브 로그의 최종 동기화를 수행합니다.
3. 원본 redo 로그를 새 서버에 복사해야 합니다. 이 예에서는 redo 로그가 의 새 디렉토리로 재배치되었습니다
/redo.

```
[oracle@jfsc3 logs]$ scp jfsc1:/logs/WAFFLE/redo/* /redo/
oracle@jfsc1's password:
redo01.log
100% 50MB 50.0MB/s 00:01
redo02.log
100% 50MB 50.0MB/s 00:00
redo03.log
100% 50MB 50.0MB/s 00:00
```

4. 이 단계에서 새 데이터베이스 환경에는 원본과 동일한 상태로 되돌리는 데 필요한 모든 파일이 포함됩니다. 아카이브 로그는 마지막으로 한 번 재생되어야 합니다.

```

SQL> recover database until cancel;
ORA-00279: change 1120099 generated at 05/26/2016 09:59:21 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_40_912662036.dbf
ORA-00280: change 1120099 for thread 1 is in sequence #40
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00308: cannot open archived log
'/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
ORA-00308: cannot open archived log
'/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3

```

5. 완료되면 재실행 로그를 재생해야 합니다. 메시지가 표시되는 경우 Media recovery complete 이 반환되고 프로세스가 성공하며 데이터베이스가 동기화되어 열 수 있습니다.

```

SQL> recover database;
Media recovery complete.
SQL> alter database open;
Database altered.

```

로그 전달 - 파일 시스템에 **ASM**을 전달합니다

이 예에서는 Oracle RMAN을 사용하여 데이터베이스를 마이그레이션하는 방법을 보여 줍니다. 이는 파일 시스템 로그 전달과 파일 시스템 로그 전달의 이전 예와 매우 유사하지만 ASM의 파일은 호스트에 표시되지 않습니다. ASM 디바이스에 있는 데이터를 마이그레이션하는 유일한 옵션은 ASM LUN을 재배치하거나 Oracle RMAN을 사용하여 복제 작업을 수행하는 것입니다.

RMAN은 Oracle ASM에서 파일을 복사하기 위한 요구 사항이지만 RMAN 사용은 ASM에 국한되지 않습니다. RMAN을 사용하여 모든 유형의 스토리지에서 다른 유형으로 마이그레이션할 수 있습니다.

이 예에서는 팬케이크라는 데이터베이스를 ASM 스토리지에서 경로의 다른 서버에 있는 일반 파일 시스템으로 재배치하는 방법을 보여 줍니다 /oradata 및 /logs.

데이터베이스 백업을 만듭니다

첫 번째 단계는 대체 서버로 마이그레이션할 데이터베이스의 백업을 만드는 것입니다. 소스가 Oracle ASM을 사용하므로 RMAN을 사용해야 합니다. 간단한 RMAN 백업은 다음과 같이 수행할 수 있습니다. 이 방법은 나중에 RMAN에서 쉽게 식별할 수 있는 태그가 지정된 백업을 생성합니다.

첫 번째 명령은 백업 대상 유형과 사용할 위치를 정의합니다. 두 번째는 데이터 파일의 백업만 시작합니다.

```

RMAN> configure channel device type disk format '/rman/pancake/%U';
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT    '/rman/pancake/%U';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT    '/rman/pancake/%U';
new RMAN configuration parameters are successfully stored
RMAN> backup database tag 'ONTAP_MIGRATION';
Starting backup at 24-MAY-16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=251 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=+ASM0/PANCAKE/system01.dbf
input datafile file number=00002 name=+ASM0/PANCAKE/sysaux01.dbf
input datafile file number=00003 name=+ASM0/PANCAKE/undotbs101.dbf
input datafile file number=00004 name=+ASM0/PANCAKE/users01.dbf
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/lgr6c161_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/lhr6c164_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16

```

백업 제어 파일

백업 제어 파일은 이 절차의 뒷부분에서 필요합니다 duplicate database 작동.

```

RMAN> backup current controlfile format '/rman/pancake/ctrl.bkp';
Starting backup at 24-MAY-16
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/ctrl.bkp tag=TAG20160524T032651 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16

```

매개 변수 파일을 백업합니다

매개 변수 파일도 새 환경에 필요합니다. 가장 간단한 방법은 현재 spfile 또는 pfile 에서 pfile 을 만드는 것입니다. 이 예제에서 원본 데이터베이스는 spfile을 사용합니다.

```

RMAN> create pfile='/rman/pancake/pfile' from spfile;
Statement processed

```

ASM 파일 이름 바꾸기 스크립트

현재 컨트롤 파일에 정의된 여러 파일 위치는 데이터베이스를 이동할 때 변경됩니다. 다음 스크립트는 프로세스를 쉽게 하기 위해 RMAN 스크립트를 생성합니다. 이 예에서는 데이터 파일 수가 매우 적은 데이터베이스를 보여 주지만 일반적으로 데이터베이스에는 수백 또는 수천 개의 데이터 파일이 포함되어 있습니다.

이 스크립트는 에서 찾을 수 있습니다 ["ASM에서 파일 시스템으로 이름 변환"](#) 그리고 이 두 가지를 수행합니다.

먼저 매개 변수를 만들어 라는 redo 로그 위치를 다시 정의합니다 log_file_name_convert. 기본적으로 대체 필드의 목록입니다. 첫 번째 필드는 현재 redo 로그의 위치이고 두 번째 필드는 새 서버의 위치입니다. 그런 다음 패턴이 반복됩니다.

두 번째 기능은 데이터 파일 이름 변경을 위한 템플릿을 제공하는 것입니다. 스크립트는 데이터 파일을 반복하고 이름 및 파일 번호 정보를 가져와서 RMAN 스크립트로 형식을 지정합니다. 그런 다음 임시 파일에서도 마찬가지입니다. 그 결과, 파일이 원하는 위치로 복구되도록 원하는 대로 편집할 수 있는 간단한 RMAN 스크립트가 생성됩니다.

```

SQL> @/rman/mk.rename.scripts.sql
Parameters for log file conversion:
*.log_file_name_convert = '+ASM0/PANCAKE/redo01.log',
'/NEW_PATH/redo01.log', '+ASM0/PANCAKE/redo02.log',
'/NEW_PATH/redo02.log', '+ASM0/PANCAKE/redo03.log', '/NEW_PATH/redo03.log'
rman duplication script:
run
{
set newname for datafile 1 to '+ASM0/PANCAKE/system01.dbf';
set newname for datafile 2 to '+ASM0/PANCAKE/sysaux01.dbf';
set newname for datafile 3 to '+ASM0/PANCAKE/undotbs101.dbf';
set newname for datafile 4 to '+ASM0/PANCAKE/users01.dbf';
set newname for tempfile 1 to '+ASM0/PANCAKE/temp01.dbf';
duplicate target database for standby backup location INSERT_PATH_HERE;
}
PL/SQL procedure successfully completed.

```

이 화면의 출력을 캡처합니다. 를 클릭합니다 log_file_name_convert 매개 변수는 아래 설명된 대로 pfile에 배치됩니다. 데이터 파일을 원하는 위치에 배치하려면 RMAN 데이터 파일 이름 바꾸기 및 중복 스크립트를 적절히 편집해야 합니다. 이 예제에서는 모두 에 배치됩니다 /oradata/pancake.

```

run
{
set newname for datafile 1 to '/oradata/pancake/pancake.dbf';
set newname for datafile 2 to '/oradata/pancake/sysaux.dbf';
set newname for datafile 3 to '/oradata/pancake/undotbs1.dbf';
set newname for datafile 4 to '/oradata/pancake/users.dbf';
set newname for tempfile 1 to '/oradata/pancake/temp.dbf';
duplicate target database for standby backup location '/rman/pancake';
}

```

디렉토리 구조를 준비합니다

스크립트는 거의 실행할 준비가 되었지만 먼저 디렉토리 구조가 있어야 합니다. 필요한 디렉터리가 아직 없으면 해당 디렉터리를 만들어야 합니다. 그렇지 않으면 데이터베이스 시작 절차가 실패합니다. 아래의 예는 최소 요구 사항을 반영합니다.

```

[oracle@jpsc2 ~]$ mkdir /oradata/pancake
[oracle@jpsc2 ~]$ mkdir /logs/pancake
[oracle@jpsc2 ~]$ cd /orabin/admin
[oracle@jpsc2 admin]$ mkdir PANCAKE
[oracle@jpsc2 admin]$ cd PANCAKE
[oracle@jpsc2 PANCAKE]$ mkdir adump dpdump pfile scripts xdb_wallet

```

ORATAB 항목을 만듭니다

oraenv와 같은 유틸리티가 제대로 작동하려면 다음 명령이 필요합니다.

```
PANCAKE:/orabin/product/12.1.0/dbhome_1:N
```

매개 변수 업데이트

새 서버의 경로 변경 사항을 반영하도록 저장된 pfile을 업데이트해야 합니다. 데이터 파일 경로 변경은 RMAN 복제 스크립트에 의해 변경되며 거의 모든 데이터베이스를 변경해야 합니다 control_files 및 log_archive_dest 매개 변수. 또한 변경해야 하는 감사 파일 위치와 같은 매개 변수가 있을 수 있습니다 db_create_file_dest ASM 외부에서는 관련이 없을 수 있습니다. 숙련된 DBA는 계속하기 전에 제안된 변경 사항을 주의 깊게 검토해야 합니다.

이 예에서 주요 변경 사항은 제어 파일 위치, 로그 아카이브 대상 및 추가입니다 log_file_name_convert 매개 변수.

```

PANCAKE.__data_transfer_cache_size=0
PANCAKE.__db_cache_size=545259520
PANCAKE.__java_pool_size=4194304
PANCAKE.__large_pool_size=25165824
PANCAKE.__oracle_base='/orabin'#ORACLE_BASE set from environment
PANCAKE.__pga_aggregate_target=268435456
PANCAKE.__sga_target=805306368
PANCAKE.__shared_io_pool_size=29360128
PANCAKE.__shared_pool_size=192937984
PANCAKE.__streams_pool_size=0
*.audit_file_dest='/orabin/admin/PANCAKE/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='+ASM0/PANCAKE/control01.ctl','+ASM0/PANCAKE/control02.ctl'
*.control_files='/oradata/pancake/control01.ctl','/logs/pancake/control02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='PANCAKE'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=PANCAKEXDB)'
*.log_archive_dest_1='LOCATION=+ASM1'
*.log_archive_dest_1='LOCATION=/logs/pancake'
*.log_archive_format='%t_%s_%r.dbf'
'/logs/path/redo02.log'
*.log_file_name_convert = '+ASM0/PANCAKE/redo01.log',
'/logs/pancake/redo01.log', '+ASM0/PANCAKE/redo02.log',
'/logs/pancake/redo02.log', '+ASM0/PANCAKE/redo03.log',
'/logs/pancake/redo03.log'
*.open_cursors=300
*.pga_aggregate_target=256m
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=768m
*.undo_tablespace='UNDOTBS1'

```

새 매개 변수가 확인되면 매개 변수가 적용되어야 합니다. 여러 옵션이 있지만 대부분의 고객은 pfile 텍스트를 기반으로 spfile을 만듭니다.

```

bash-4.1$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Fri Jan 8 11:17:40 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to an idle instance.
SQL> create spfile from pfile='/rman/pancake/pfile';
File created.

```

시동 nomount

데이터베이스를 복제하기 전의 마지막 단계는 데이터베이스 프로세스를 불러오지만 파일을 마운트하지 않는 것입니다. 이 단계에서는 spfile에 문제가 발생할 수 있습니다. 를 누릅니다 startup nomount 명령 실패 매개 변수 오류로 인해 실패합니다. pfile 템플릿을 종료하고 수정한 다음 spfile로 다시 로드한 후 다시 시도하십시오.

```

SQL> startup nomount;
ORACLE instance started.
Total System Global Area 805306368 bytes
Fixed Size 2929552 bytes
Variable Size 373296240 bytes
Database Buffers 423624704 bytes
Redo Buffers 5455872 bytes

```

데이터베이스를 복제합니다

이전 RMAN 백업을 새 위치로 복원하는 데 이 프로세스의 다른 단계보다 시간이 더 오래 걸립니다. 데이터베이스 ID(DBID)를 변경하거나 로그를 재설정하지 않고 데이터베이스를 복제해야 합니다. 이렇게 하면 로그를 적용할 수 없습니다. 이는 복사본을 완전히 동기화하는 데 필요한 단계입니다.

RMAN을 aux로 데이터베이스에 연결하고 이전 단계에서 생성한 스크립트를 사용하여 중복 데이터베이스 명령을 실행합니다.

```

[oracle@jfsc2 pancake]$ rman auxiliary /
Recovery Manager: Release 12.1.0.2.0 - Production on Tue May 24 03:04:56
2016
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights
reserved.
connected to auxiliary database: PANCAKE (not mounted)
RMAN> run
2> {
3> set newname for datafile 1 to '/oradata/pancake/pancake.dbf';
4> set newname for datafile 2 to '/oradata/pancake/sysaux.dbf';
5> set newname for datafile 3 to '/oradata/pancake/undotbs1.dbf';
6> set newname for datafile 4 to '/oradata/pancake/users.dbf';
7> set newname for tempfile 1 to '/oradata/pancake/temp.dbf';
8> duplicate target database for standby backup location '/rman/pancake';
9> }

```

```

executing command: SET NEWNAME
Starting Duplicate Db at 24-MAY-16
contents of Memory Script:
{
    restore clone standby controlfile from  '/rman/pancake/ctrl.bkp';
}
executing Memory Script
Starting restore at 24-MAY-16
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=243 device type=DISK
channel ORA_AUX_DISK_1: restoring control file
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
output file name=/oradata/pancake/control01.ctl
output file name=/logs/pancake/control02.ctl
Finished restore at 24-MAY-16
contents of Memory Script:
{
    sql clone 'alter database mount standby database';
}
executing Memory Script
sql statement: alter database mount standby database
released channel: ORA_AUX_DISK_1
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=243 device type=DISK
contents of Memory Script:
{
    set newname for tempfile 1 to
"/oradata/pancake/temp.dbf";
    switch clone tempfile all;
    set newname for datafile 1 to
"/oradata/pancake/pancake.dbf";
    set newname for datafile 2 to
"/oradata/pancake/sysaux.dbf";
    set newname for datafile 3 to
"/oradata/pancake/undotbs1.dbf";
    set newname for datafile 4 to
"/oradata/pancake/users.dbf";
    restore
    clone database
    ;
}
executing Memory Script

```

```
executing command: SET NEWNAME
renamed tempfile 1 to /oradata/pancake/temp.dbf in control file
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
Starting restore at 24-MAY-16
using channel ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_AUX_DISK_1: restoring datafile 00001 to
/oradata/pancake/pancake.dbf
channel ORA_AUX_DISK_1: restoring datafile 00002 to
/oradata/pancake/sysaux.dbf
channel ORA_AUX_DISK_1: restoring datafile 00003 to
/oradata/pancake/undotbs1.dbf
channel ORA_AUX_DISK_1: restoring datafile 00004 to
/oradata/pancake/users.dbf
channel ORA_AUX_DISK_1: reading from backup piece
/rman/pancake/1gr6c161_1_1
channel ORA_AUX_DISK_1: piece handle=/rman/pancake/1gr6c161_1_1
tag=ONTAP_MIGRATION
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:07
Finished restore at 24-MAY-16
contents of Memory Script:
{
    switch clone datafile all;
}
executing Memory Script
datafile 1 switched to datafile copy
input datafile copy RECID=5 STAMP=912655725 file
name=/oradata/pancake/pancake.dbf
datafile 2 switched to datafile copy
input datafile copy RECID=6 STAMP=912655725 file
name=/oradata/pancake/sysaux.dbf
datafile 3 switched to datafile copy
input datafile copy RECID=7 STAMP=912655725 file
name=/oradata/pancake/undotbs1.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=8 STAMP=912655725 file
name=/oradata/pancake/users.dbf
Finished Duplicate Db at 24-MAY-16
```

이제 원본 데이터베이스의 변경 내용을 새 위치로 전달해야 합니다. 이렇게 하려면 여러 단계를 조합해야 할 수 있습니다. 가장 간단한 방법은 소스 데이터베이스의 RMAN이 공유 네트워크 연결에 아카이브 로그를 기록하도록 하는 것입니다. 공유 위치를 사용할 수 없는 경우 RMAN을 사용하여 로컬 파일 시스템에 쓴 다음 RCP 또는 rsync를 사용하여 파일을 복사하는 방법이 있습니다.

이 예에서 는 입니다 /rman 디렉토리는 원래 데이터베이스와 마이그레이션된 데이터베이스 모두에서 사용할 수 있는 NFS 공유입니다.

여기서 한 가지 중요한 문제는 입니다 disk format 조항. 백업의 디스크 형식은 입니다 %h_%e_%a.dbf`즉, 데이터베이스에 대한 스레드 번호, 시퀀스 번호 및 활성화 ID 형식을 사용해야 합니다. 글자는 다르지만 이 문장은 과 일치합니다 `log_archive_format='%t_%s_%r.dbf pfile의 매개 변수입니다. 또한 이 매개 변수는 스레드 번호, 시퀀스 번호 및 활성화 ID 형식으로 아카이브 로그를 지정합니다. 결과적으로 소스의 로그 파일 백업이 데이터베이스에서 예상하는 명명 규칙을 사용하게 됩니다. 이렇게 하면 과 같은 작업이 수행됩니다 recover database sqlplus 는 재생될 아카이브 로그의 이름을 올바르게 예측하기 때문에 훨씬 더 간단합니다.

```

RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/arch/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
released channel: ORA_DISK_1
RMAN> backup as copy archivelog from time 'sysdate-2';
Starting backup at 24-MAY-16
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=373 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=70 STAMP=912658508
output file name=/rman/pancake/logship/1_54_912576125.dbf RECID=123
STAMP=912659482
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=41 RECID=29 STAMP=912654101
output file name=/rman/pancake/logship/1_41_912576125.dbf RECID=124
STAMP=912659483
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
...
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=45 RECID=33 STAMP=912654688
output file name=/rman/pancake/logship/1_45_912576125.dbf RECID=152
STAMP=912659514
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=36 STAMP=912654809
output file name=/rman/pancake/logship/1_47_912576125.dbf RECID=153
STAMP=912659515
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16

```

초기 로그 재생

파일이 아카이브 로그 위치에 있으면 명령을 실행하여 재생할 수 있습니다 `recover database until cancel` 그 다음에 응답이 옵니다 `AUTO` 사용 가능한 모든 로그를 자동으로 재생합니다. 매개 변수 파일이 현재 아카이브 로그를 로 리디렉션하고 있습니다 ``logs/archive`` 하지만 RMAN이 로그를 저장하는 데 사용된 위치와 일치하지 않습니다. 데이터베이스를 복구하기 전에 다음과 같이 위치를 일시적으로 리디렉션할 수 있습니다.

```

SQL> alter system set log_archive_dest_1='LOCATION=/rman/pancake/logship'
scope=memory;
System altered.
SQL> recover standby database until cancel;
ORA-00279: change 560224 generated at 05/24/2016 03:25:53 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_49_912576125.dbf
ORA-00280: change 560224 for thread 1 is in sequence #49
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00279: change 560353 generated at 05/24/2016 03:29:17 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_50_912576125.dbf
ORA-00280: change 560353 for thread 1 is in sequence #50
ORA-00278: log file '/rman/pancake/logship/1_49_912576125.dbf' no longer
needed
for this recovery
...
ORA-00279: change 560591 generated at 05/24/2016 03:33:56 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_54_912576125.dbf
ORA-00280: change 560591 for thread 1 is in sequence #54
ORA-00278: log file '/rman/pancake/logship/1_53_912576125.dbf' no longer
needed
for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_54_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3

```

최종 아카이브 로그 응답에서 오류를 보고하지만 이는 정상입니다. 이 오류는 sqlplus가 특정 로그 파일을 찾고 있지만 찾지 못했음을 나타냅니다. 로그 파일이 아직 존재하지 않기 때문일 수 있습니다.

아카이브 로그를 복사하기 전에 소스 데이터베이스를 종료할 수 있는 경우 이 단계는 한 번만 수행해야 합니다. 아카이브 로그가 복사되고 재생된 다음 프로세스를 계속 진행하여 중요한 재실행 로그를 복제하는 컷오버 프로세스로 이동할 수 있습니다.

증분 로그 복제 및 재생

대부분의 경우 마이그레이션은 즉시 수행되지 않습니다. 마이그레이션 프로세스가 완료되기까지 며칠이나 몇 주가 걸릴 수 있습니다. 즉, 로그가 계속해서 복제본 데이터베이스로 전송되고 재생되어야 합니다. 이렇게 하면 컷오버가 도착할 때 최소한의 데이터를 전송하고 재생해야 합니다.

이 프로세스는 쉽게 스크립팅할 수 있습니다. 예를 들어, 로그 전달에 사용되는 위치가 지속적으로 업데이트되도록 원본 데이터베이스에 다음 명령을 예약할 수 있습니다.

```
[oracle@jfscl pancake]$ cat copylogs.rman
configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
backup as copy archivelog from time 'sysdate-2';
```

```
[oracle@jfscl pancake]$ rman target / cmdfile=copylogs.rman
Recovery Manager: Release 12.1.0.2.0 - Production on Tue May 24 04:36:19
2016
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights
reserved.
connected to target database: PANCAKE (DBID=3574534589)
RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
2> backup as copy archivelog from time 'sysdate-2';
3>
4>
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
Starting backup at 24-MAY-16
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=369 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=123 STAMP=912659482
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:22
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_54_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=41 RECID=124 STAMP=912659483
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:23
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_41_912576125.dbf
continuing other job steps, job failed will not be re-run
...
channel ORA_DISK_1: starting archived log copy
```

```
input archived log thread=1 sequence=45 RECID=152 STAMP=912659514
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:55
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_45_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=153 STAMP=912659515
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:57
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_47_912576125.dbf
Recovery Manager complete.
```

로그를 수신한 후 재생해야 합니다. 이전 예제에서는 sqlplus 를 사용하여 수동으로 실행하는 방법을 보여 주었습니다 `recover database until cancel` 쉽게 자동화할 수 있습니다. 여기에 표시된 예에서는 에 설명된 스크립트를 사용합니다 "대기 데이터베이스에서 로그를 재생합니다". 스크립트에는 재생 작업이 필요한 데이터베이스를 지정하는 인수를 사용할 수 있습니다. 이 프로세스에서는 다중 데이터베이스 마이그레이션 작업에 동일한 스크립트를 사용할 수 있습니다.

```
[root@jpsc2 pancake]# ./replaylogs.pl PANCAKE
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 24 04:47:10 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 560591 generated at 05/24/2016 03:33:56 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_54_912576125.dbf
ORA-00280: change 560591 for thread 1 is in sequence #54
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 562219 generated at 05/24/2016 04:15:08 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_55_912576125.dbf
ORA-00280: change 562219 for thread 1 is in sequence #55
ORA-00278: log file '/rman/pancake/logship/1_54_912576125.dbf' no longer
needed for this recovery
ORA-00279: change 562370 generated at 05/24/2016 04:19:18 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_56_912576125.dbf
ORA-00280: change 562370 for thread 1 is in sequence #56
ORA-00278: log file '/rman/pancake/logship/1_55_912576125.dbf' no longer
needed for this recovery
...
ORA-00279: change 563137 generated at 05/24/2016 04:36:20 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_65_912576125.dbf
ORA-00280: change 563137 for thread 1 is in sequence #65
ORA-00278: log file '/rman/pancake/logship/1_64_912576125.dbf' no longer
needed for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_65_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
```

컷오버

새 환경으로 컷오버할 준비가 되면 최종 동기화 하나를 수행해야 합니다. 일반 파일 시스템으로 작업할 때 원래 redo 로그가 복사되고 재생되므로 마이그레이션된 데이터베이스가 원본과 100% 동기화되도록 쉽게 할 수 있습니다. ASM과 함께 이 작업을 수행하는 좋은 방법은 없습니다. 보관 로그만 쉽게 다시 복사할 수 있습니다. 데이터가 손실되지 않도록 하려면 원본 데이터베이스의 최종 종료로 주의 깊게 수행해야 합니다.

1. 먼저 데이터베이스를 정지하여 변경 사항이 없는지 확인해야 합니다. 이 일시 중지에는 예약된 작업을 비활성화하거나, 수신키를 종료하거나, 응용 프로그램을 종료하는 작업이 포함될 수 있습니다.
2. 이 단계를 수행한 후 대부분의 DBA는 종료의 표시자 역할을 하는 더미 테이블을 생성합니다.
3. 로그 아카이빙을 강제 수행하여 더미 테이블 생성이 아카이브 로그 내에 기록되도록 합니다. 이렇게 하려면 다음 명령을 실행합니다.

```
SQL> create table cutovercheck as select * from dba_users;
Table created.
SQL> alter system archive log current;
System altered.
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
```

4. 마지막 아카이브 로그를 복사하려면 다음 명령을 실행합니다. 데이터베이스를 사용할 수 있어야 하지만 열려 있지 않아야 합니다.

```
SQL> startup mount;
ORACLE instance started.
Total System Global Area 805306368 bytes
Fixed Size 2929552 bytes
Variable Size 331353200 bytes
Database Buffers 465567744 bytes
Redo Buffers 5455872 bytes
Database mounted.
```

5. 아카이브 로그를 복사하려면 다음 명령을 실행합니다.

```

RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
2> backup as copy archivelog from time 'sysdate-2';
3>
4>
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
Starting backup at 24-MAY-16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=8 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=123 STAMP=912659482
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:58:24
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_54_912576125.dbf
continuing other job steps, job failed will not be re-run
...
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=45 RECID=152 STAMP=912659514
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:58:58
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_45_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=153 STAMP=912659515
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:59:00
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_47_912576125.dbf

```

6. 마지막으로 새 서버에서 나머지 아카이브 로그를 재생합니다.

```

[root@jpsc2 pancake]# ./replaylogs.pl PANCAKE
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 24 05:00:53 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 563137 generated at 05/24/2016 04:36:20 needed
for thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_65_912576125.dbf
ORA-00280: change 563137 for thread 1 is in sequence #65
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 563629 generated at 05/24/2016 04:55:20 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_66_912576125.dbf
ORA-00280: change 563629 for thread 1 is in sequence #66
ORA-00278: log file '/rman/pancake/logship/1_65_912576125.dbf' no longer
needed
for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_66_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options

```

7. 이 단계에서는 모든 데이터를 복제합니다. 데이터베이스를 대기 데이터베이스에서 활성 작업 데이터베이스로 변환할 준비가 된 다음 열 수 있습니다.

```

SQL> alter database activate standby database;
Database altered.
SQL> alter database open;
Database altered.

```

8. 더미 테이블이 있는지 확인한 다음 삭제합니다.

```

SQL> desc cutovercheck
Name                                                    Null?    Type
-----
-----
USERNAME                                                NOT NULL VARCHAR2 (128)
USER_ID                                                  NOT NULL NUMBER
PASSWORD                                                VARCHAR2 (4000)
ACCOUNT_STATUS                                          NOT NULL VARCHAR2 (32)
LOCK_DATE                                               DATE
EXPIRY_DATE                                             DATE
DEFAULT_TABLESPACE                                     NOT NULL VARCHAR2 (30)
TEMPORARY_TABLESPACE                                  NOT NULL VARCHAR2 (30)
CREATED                                                 NOT NULL DATE
PROFILE                                                 NOT NULL VARCHAR2 (128)
INITIAL_RSRC_CONSUMER_GROUP                            VARCHAR2 (128)
EXTERNAL_NAME                                           VARCHAR2 (4000)
PASSWORD_VERSIONS                                       VARCHAR2 (12)
EDITIONS_ENABLED                                       VARCHAR2 (1)
AUTHENTICATION_TYPE                                    VARCHAR2 (8)
PROXY_ONLY_CONNECT                                    VARCHAR2 (1)
COMMON                                                  VARCHAR2 (3)
LAST_LOGIN                                              TIMESTAMP (9) WITH
TIME_ZONE
ORACLE_MAINTAINED                                       VARCHAR2 (1)
SQL> drop table cutovercheck;
Table dropped.

```

무중단 재실행 로그 마이그레이션

재실행 로그를 제외하고 데이터베이스가 전체적으로 올바르게 구성된 경우가 있습니다. 이러한 현상은 여러 가지 이유로 발생할 수 있으며, 그 중 가장 일반적인 원인은 스냅샷과 관련이 있습니다. Oracle용 SnapManager, SnapCenter, NetApp Snap Creator 스토리지 관리 프레임워크와 같은 제품을 사용하면 데이터 파일 볼륨의 상태를 되돌리는 경우에만 거의 즉각적으로 데이터베이스 복구가 가능합니다. 재실행 로그가 데이터 파일과 공간을 공유하는 경우 재실행 로그가 삭제되어 데이터 손실이 발생할 수 있으므로 재버전을 안전하게 수행할 수 없습니다. 따라서 redo 로그를 재배치해야 합니다.

이 절차는 단순하며 중단 없이 수행할 수 있습니다.

현재 redo 로그 구성

1. 재실행 로그 그룹의 수와 해당 그룹 번호를 식별합니다.

```

SQL> select group#||' '||member from v$logfile;
GROUP#||' '||MEMBER
-----
-----
1 /redo0/NTAP/redo01a.log
1 /redo1/NTAP/redo01b.log
2 /redo0/NTAP/redo02a.log
2 /redo1/NTAP/redo02b.log
3 /redo0/NTAP/redo03a.log
3 /redo1/NTAP/redo03b.log
rows selected.

```

2. redo 로그의 크기를 입력합니다.

```

SQL> select group#||' '||bytes from v$log;
GROUP#||' '||BYTES
-----
-----
1 524288000
2 524288000
3 524288000

```

새 로그를 만듭니다

1. 각 REDO 로그에 대해 일치하는 크기와 구성원 수가 있는 새 그룹을 만듭니다.

```

SQL> alter database add logfile ('/newredo0/redo01a.log',
'/newredo1/redo01b.log') size 500M;
Database altered.
SQL> alter database add logfile ('/newredo0/redo02a.log',
'/newredo1/redo02b.log') size 500M;
Database altered.
SQL> alter database add logfile ('/newredo0/redo03a.log',
'/newredo1/redo03b.log') size 500M;
Database altered.
SQL>

```

2. 새 구성을 확인합니다.

```

SQL> select group#||' '||member from v$logfile;
GROUP#||' '||MEMBER
-----
-----
1 /redo0/NTAP/redo01a.log
1 /redo1/NTAP/redo01b.log
2 /redo0/NTAP/redo02a.log
2 /redo1/NTAP/redo02b.log
3 /redo0/NTAP/redo03a.log
3 /redo1/NTAP/redo03b.log
4 /newredo0/redo01a.log
4 /newredo1/redo01b.log
5 /newredo0/redo02a.log
5 /newredo1/redo02b.log
6 /newredo0/redo03a.log
6 /newredo1/redo03b.log
12 rows selected.

```

오래된 로그를 삭제합니다

1. 이전 로그(그룹 1, 2, 3)를 삭제합니다.

```

SQL> alter database drop logfile group 1;
Database altered.
SQL> alter database drop logfile group 2;
Database altered.
SQL> alter database drop logfile group 3;
Database altered.

```

2. 활성 로그를 삭제할 수 없는 오류가 발생하면 다음 로그로 스위치를 강제로 전환하여 잠금을 해제하고 글로벌 체크포인트를 강제로 설정합니다. 이 프로세스의 다음 예를 참조하십시오. 이 로그 파일에 활성 데이터가 있기 때문에 이전 위치에 있던 로그 파일 그룹 2를 삭제하려는 시도가 거부되었습니다.

```

SQL> alter database drop logfile group 2;
alter database drop logfile group 2
*
ERROR at line 1:
ORA-01623: log 2 is current log for instance NTAP (thread 1) - cannot
drop
ORA-00312: online log 2 thread 1: '/redo0/NTAP/redo02a.log'
ORA-00312: online log 2 thread 1: '/redo1/NTAP/redo02b.log'

```

3. 로그 보관 후 체크포인트를 수행하면 로그 파일을 삭제할 수 있습니다.

```
SQL> alter system archive log current;
System altered.
SQL> alter system checkpoint;
System altered.
SQL> alter database drop logfile group 2;
Database altered.
```

4. 그런 다음 파일 시스템에서 로그를 삭제합니다. 이 과정은 매우 세심한 주의를 기울여 수행해야 합니다.

Oracle 데이터베이스 호스트 데이터 복사본

데이터베이스 레벨 마이그레이션과 마찬가지로 호스트 계층에서 마이그레이션하면 스토리지 공급업체에 종속되지 않는 접근 방식이 제공됩니다.

다시 말해, 언젠가는 "그냥 파일 복사"가 가장 좋은 옵션입니다.

이러한 낮은 수준의 기술 접근 방식은 매우 기본적이라고 생각될 수 있지만, 특별한 소프트웨어가 필요하지 않고 원본 데이터가 프로세스 중에 안전하게 유지되기 때문에 상당한 이점을 제공합니다. 주된 제한 사항은 파일 복사 데이터 마이그레이션은 복사 작업을 시작하기 전에 데이터베이스를 종료해야 하기 때문에 중단 프로세스라는 점입니다. 파일 내의 변경 내용을 동기화하는 좋은 방법은 없으므로 복사하기 전에 파일을 완전히 정지해야 합니다.

복사 작업에 필요한 종료가 바람직하지 않은 경우 다음으로 가장 좋은 호스트 기반 옵션은 논리 볼륨 관리자(LVM)를 활용하는 것입니다. Oracle ASM을 비롯한 많은 LVM 옵션이 존재하며 모든 기능이 비슷하지만 몇 가지 제한도 고려해야 합니다. 대부분의 경우 다운타임 및 운영 중단 없이 마이그레이션을 수행할 수 있습니다.

파일 시스템 대 파일 시스템 복제

단순 복사 작업의 유용성을 과소 평가해서는 안 됩니다. 이 작업은 복사 프로세스 중에 다운타임이 발생하지만 매우 안정적인 프로세스이며 운영 체제, 데이터베이스 또는 스토리지 시스템에 대한 특별한 전문 지식이 필요하지 않습니다. 또한 원본 데이터에 영향을 주지 않기 때문에 매우 안전합니다. 일반적으로 시스템 관리자는 소스 파일 시스템을 읽기 전용으로 마운트하도록 변경한 다음 서버를 재부팅하여 현재 데이터가 손상되지 않도록 합니다. 복제 프로세스는 사용자 오류의 위험 없이 가능한 한 빨리 실행되도록 스크립트될 수 있습니다. I/O 유형은 데이터의 단순한 순차 전송이므로 대역폭 효율성이 매우 높습니다.

다음 예에서는 안전하고 신속한 마이그레이션을 위한 한 가지 옵션을 보여 줍니다.

방법입니다

마이그레이션할 환경은 다음과 같습니다.

- 현재 파일 시스템

```
ontap-nfs1:/host1_oradata          52428800  16196928  36231872  31%
/oradata
ontap-nfs1:/host1_logs            49807360   548032  49259328  2% /logs
```

- 새 파일 시스템

```

ontap-nfs1:/host1_logs_new      49807360      128  49807232    1%
/new/logs
ontap-nfs1:/host1_oradata_new   49807360      128  49807232    1%
/new/oradata

```

개요

데이터베이스를 종료하고 파일을 복사하기만 하면 DBA가 데이터베이스를 마이그레이션할 수 있지만, 많은 데이터베이스를 마이그레이션해야 하거나 가동 중지 시간을 최소화하는 것이 중요한 경우 프로세스가 쉽게 스크립팅됩니다. 또한 스크립트를 사용하면 사용자의 실수 가능성이 줄어듭니다.

표시된 예제 스크립트는 다음 작업을 자동화합니다.

- 데이터베이스를 종료합니다
- 기존 파일 시스템을 읽기 전용 상태로 변환합니다
- 소스의 모든 데이터를 타겟 파일 시스템으로 복제하여 모든 파일 사용 권한을 유지합니다
- 이전 파일 시스템과 새 파일 시스템의 마운트를 해제합니다
- 이전 파일 시스템과 동일한 경로에 새 파일 시스템을 다시 마운트합니다

절차를 참조하십시오

1. 데이터베이스를 종료합니다.

```

[root@host1 current]# ./dbshut.pl NTAP
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 15:58:48 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP shut down

```

2. 파일 시스템을 읽기 전용으로 변환합니다. 예서와 같이 스크립트를 사용하여 보다 빠르게 수행할 수 있습니다 "파일 시스템을 읽기 전용으로 변환합니다".

```
[root@host1 current]# ./mk.fs.readonly.pl /oradata
/oradata unmounted
/oradata mounted read-only
[root@host1 current]# ./mk.fs.readonly.pl /logs
/logs unmounted
/logs mounted read-only
```

3. 이제 파일 시스템이 읽기 전용인지 확인합니다.

```
ontap-nfs1:/host1_oradata on /oradata type nfs
(ro,bg,vers=3,rsize=65536,wsiz=65536,addr=172.20.101.10)
ontap-nfs1:/host1_logs on /logs type nfs
(ro,bg,vers=3,rsize=65536,wsiz=65536,addr=172.20.101.10)
```

4. 와 파일 시스템 콘텐츠를 동기화합니다 rsync 명령.

```
[root@host1 current]# rsync -rlpogt --stats --progress
--exclude=.snapshot /oradata/ /new/oradata/
sending incremental file list
./
NTAP/
NTAP/IOPS.dbf
 10737426432 100% 153.50MB/s   0:01:06 (xfer#1, to-check=10/13)
NTAP/iops.dbf.zip
  22823573 100%  12.09MB/s   0:00:01 (xfer#2, to-check=9/13)
...
NTAP/undotbs02.dbf
 1073750016 100% 131.60MB/s   0:00:07 (xfer#10, to-check=1/13)
NTAP/users01.dbf
  5251072 100%   3.95MB/s   0:00:01 (xfer#11, to-check=0/13)
Number of files: 13
Number of files transferred: 11
Total file size: 18570092218 bytes
Total transferred file size: 18570092218 bytes
Literal data: 18570092218 bytes
Matched data: 0 bytes
File list size: 277
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 18572359828
Total bytes received: 228
sent 18572359828 bytes  received 228 bytes  162204017.96 bytes/sec
total size is 18570092218  speedup is 1.00
```

```

[root@host1 current]# rsync -rlpogt --stats --progress
--exclude=.snapshot /logs/ /new/logs/
sending incremental file list
./
NTAP/
NTAP/1_22_897068759.dbf
    45523968 100%  95.98MB/s    0:00:00 (xfer#1, to-check=15/18)
NTAP/1_23_897068759.dbf
    40601088 100%  49.45MB/s    0:00:00 (xfer#2, to-check=14/18)
...
NTAP/redo/redo02.log
    52429312 100%  44.68MB/s    0:00:01 (xfer#12, to-check=1/18)
NTAP/redo/redo03.log
    52429312 100%  68.03MB/s    0:00:00 (xfer#13, to-check=0/18)
Number of files: 18
Number of files transferred: 13
Total file size: 527032832 bytes
Total transferred file size: 527032832 bytes
Literal data: 527032832 bytes
Matched data: 0 bytes
File list size: 413
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 527098156
Total bytes received: 278
sent 527098156 bytes  received 278 bytes  95836078.91 bytes/sec
total size is 527032832  speedup is 1.00

```

5. 이전 파일 시스템을 마운트 해제하고 복제된 데이터를 재배치합니다. 예서와 같이 스크립트를 사용하여 보다 빠르게 수행할 수 있습니다 **"파일 시스템을 교체합니다"**.

```

[root@host1 current]# ./swap.fs.pl /logs,/new/logs
/new/logs unmounted
/logs unmounted
Updated /logs mounted
[root@host1 current]# ./swap.fs.pl /oradata,/new/oradata
/new/oradata unmounted
/oradata unmounted
Updated /oradata mounted

```

6. 새 파일 시스템이 올바른 위치에 있는지 확인합니다.

```
ontap-nfs1:/host1_logs_new on /logs type nfs
(rw,bg,vers=3,rsz=65536,wsz=65536,addr=172.20.101.10)
ontap-nfs1:/host1_oradata_new on /oradata type nfs
(rw,bg,vers=3,rsz=65536,wsz=65536,addr=172.20.101.10)
```

7. 데이터베이스를 시작합니다.

```
[root@host1 current]# ./dbstart.pl NTAP
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 16:10:07 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to an idle instance.
SQL> ORACLE instance started.
Total System Global Area 805306368 bytes
Fixed Size 2929552 bytes
Variable Size 390073456 bytes
Database Buffers 406847488 bytes
Redo Buffers 5455872 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP started
```

완전 자동화된 컷오버

이 샘플 스크립트에서는 데이터베이스 SID의 인수 다음에 파일 시스템의 공용 구분 쌍이 오는 인수를 사용할 수 있습니다. 위의 예에서 명령은 다음과 같이 실행됩니다.

```
[root@host1 current]# ./migrate.oracle.fs.pl NTAP /logs,/new/logs
/oradata,/new/oradata
```

예제 스크립트가 실행되면 다음 순서를 수행하려고 시도합니다. 어떤 단계에서든 오류가 발생하면 종료됩니다.

1. 데이터베이스를 종료합니다.
2. 현재 파일 시스템을 읽기 전용 상태로 변환합니다.
3. 심표로 구분된 각 파일 시스템 인수 쌍을 사용하고 첫 번째 파일 시스템을 두 번째 파일 시스템과 동기화합니다.
4. 이전 파일 시스템을 분리합니다.
5. 를 업데이트합니다 /etc/fstab 다음과 같은 파일:

- a. 에서 백업을 생성합니다 /etc/fstab.bak.
 - b. 이전 및 새 파일 시스템에 대한 이전 항목을 주석 처리합니다.
 - c. 이전 마운트 지점을 사용하는 새 파일 시스템에 대한 새 항목을 생성합니다.
6. 파일 시스템을 마운트합니다.
7. 데이터베이스를 시작합니다.

다음 텍스트는 이 스크립트에 대한 실행 예제를 제공합니다.

```
[root@host1 current]# ./migrate.oracle.fs.pl NTAP /logs,/new/logs
/oradata,/new/oradata
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 17:05:50 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP shut down
sending incremental file list
./
NTAP/
NTAP/1_22_897068759.dbf
      45523968 100%  185.40MB/s   0:00:00 (xfer#1, to-check=15/18)
NTAP/1_23_897068759.dbf
      40601088 100%   81.34MB/s   0:00:00 (xfer#2, to-check=14/18)
...
NTAP/redo/redo02.log
      52429312 100%   70.42MB/s   0:00:00 (xfer#12, to-check=1/18)
NTAP/redo/redo03.log
      52429312 100%   47.08MB/s   0:00:01 (xfer#13, to-check=0/18)
Number of files: 18
Number of files transferred: 13
Total file size: 527032832 bytes
Total transferred file size: 527032832 bytes
Literal data: 527032832 bytes
Matched data: 0 bytes
```

```

File list size: 413
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 527098156
Total bytes received: 278
sent 527098156 bytes received 278 bytes 150599552.57 bytes/sec
total size is 527032832 speedup is 1.00
Succesfully replicated filesystem /logs to /new/logs
sending incremental file list
./
NTAP/
NTAP/IOPS.dbf
  10737426432 100% 176.55MB/s 0:00:58 (xfer#1, to-check=10/13)
NTAP/iops.dbf.zip
  22823573 100% 9.48MB/s 0:00:02 (xfer#2, to-check=9/13)
... NTAP/undotbs01.dbf
  309338112 100% 70.76MB/s 0:00:04 (xfer#9, to-check=2/13)
NTAP/undotbs02.dbf
  1073750016 100% 187.65MB/s 0:00:05 (xfer#10, to-check=1/13)
NTAP/users01.dbf
  5251072 100% 5.09MB/s 0:00:00 (xfer#11, to-check=0/13)
Number of files: 13
Number of files transferred: 11
Total file size: 18570092218 bytes
Total transferred file size: 18570092218 bytes
Literal data: 18570092218 bytes
Matched data: 0 bytes
File list size: 277
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 18572359828
Total bytes received: 228
sent 18572359828 bytes received 228 bytes 177725933.55 bytes/sec
total size is 18570092218 speedup is 1.00
Succesfully replicated filesystem /oradata to /new/oradata
swap 0 /logs /new/logs
/new/logs unmounted
/logs unmounted
Mounted updated /logs
Swapped filesystem /logs for /new/logs
swap 1 /oradata /new/oradata
/new/oradata unmounted
/oradata unmounted
Mounted updated /oradata
Swapped filesystem /oradata for /new/oradata
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin

```

```

SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 17:08:59 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to an idle instance.
SQL> ORACLE instance started.
Total System Global Area 805306368 bytes
Fixed Size 2929552 bytes
Variable Size 390073456 bytes
Database Buffers 406847488 bytes
Redo Buffers 5455872 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP started
[root@host1 current]#

```

Oracle ASM spfile 및 passwd 마이그레이션

ASM과 관련된 마이그레이션을 완료하는 데 있어 한 가지 어려움은 ASM 관련 spfile과 암호 파일입니다. 기본적으로 이러한 중요 메타데이터 파일은 정의된 첫 번째 ASM 디스크 그룹에 생성됩니다. 특정 ASM 디스크 그룹을 비우고 제거해야 하는 경우 해당 ASM 인스턴스를 제어하는 spfile 및 암호 파일을 재배치해야 합니다.

이러한 파일을 재배치해야 하는 또 다른 활용 사례는 SnapManager for Oracle 또는 SnapCenter Oracle 플러그인과 같은 데이터베이스 관리 소프트웨어를 구축하는 경우입니다. 이러한 제품의 기능 중 하나는 데이터 파일을 호스팅하는 ASM LUN의 상태를 되돌려 신속하게 데이터베이스를 복원하는 것입니다. 이렇게 하려면 복원을 수행하기 전에 ASM 디스크 그룹을 오프라인으로 전환해야 합니다. 특정 데이터베이스의 데이터 파일이 전용 ASM 디스크 그룹에서 격리되어 있는 한 이 문제는 문제가 되지 않습니다.

해당 디스크 그룹에 ASM spfile/passwd 파일도 포함되어 있는 경우 디스크 그룹을 오프라인으로 전환할 수 있는 유일한 방법은 전체 ASM 인스턴스를 종료하는 것입니다. 이는 중단 프로세스이므로 spfile/passwd 파일을 재배치해야 합니다.

방법입니다

1. 데이터베이스 SID = 토스트
2. 의 현재 데이터 파일 +DATA
3. 의 현재 로그 파일 및 제어 파일 +LOGS
4. 로 설정된 새 ASM 디스크 그룹 +NEWDATA 및 +NEWLOGS

ASM spfile/passwd 파일 위치

이러한 파일 재배치는 중단 없이 수행할 수 있습니다. 그러나 안전을 위해 NetApp에서는 파일이 재배치되고 구성이 올바르게 업데이트되었는지 확인할 수 있도록 데이터베이스 환경을 종료하는 것이 좋습니다. 서버에 여러 ASM 인스턴스가 있는 경우 이 절차를 반복해야 합니다.

ASM 인스턴스 식별

에 기록된 데이터를 기반으로 ASM 인스턴스를 식별합니다 oratab 파일. ASM 인스턴스는 + 기호로 표시됩니다.

```
-bash-4.1$ cat /etc/oratab | grep '^+'
+ASM:/orabin/grid:N          # line added by Agent
```

이 서버에는 +asm 이라는 ASM 인스턴스가 하나 있습니다.

모든 데이터베이스가 종료되었는지 확인합니다

사용 중인 ASM 인스턴스에 대한 smon 프로세스만 볼 수 있습니다. 다른 스몬 프로세스가 있으면 데이터베이스가 여전히 실행 중임을 나타냅니다.

```
-bash-4.1$ ps -ef | grep smon
oracle      857      1  0 18:26 ?          00:00:00 asm_smon_+ASM
```

유일한 smon 프로세스는 ASM 인스턴스 자체입니다. 즉, 다른 데이터베이스는 실행 중이지 않으므로 데이터베이스 작업을 중단하지 않고 계속 진행하는 것이 안전합니다.

파일을 찾습니다

를 사용하여 ASM spfile 및 암호 파일의 현재 위치를 식별합니다 spget 및 pwget 명령.

```
bash-4.1$ asmcmd
ASMCMD> spget
+DATA/spfile.ora
```

```
ASMCMD> pwget --asm
+DATA/orapwasm
```

두 파일은 모두 의 하단에 있습니다 +DATA 디스크 그룹입니다.

파일을 복사합니다

를 사용하여 파일을 새 ASM 디스크 그룹에 복사합니다 spcopy 및 pwcop 명령. 새 디스크 그룹이 최근에 생성되어 현재 비어 있는 경우 먼저 마운트해야 할 수 있습니다.

```
ASMCMD> mount NEWDATA
```

```
ASMCMD> spcopy +DATA/spfile.ora +NEWDATA/spfile.ora
copying +DATA/spfile.ora -> +NEWDATA/spfilea.ora
```

```
ASMCMD> pwcopy +DATA/orapwasm +NEWDATA/orapwasm
copying +DATA/orapwasm -> +NEWDATA/orapwasm
```

이제 파일이 에서 복사되었습니다 +DATA 를 선택합니다 +NEWDATA.

ASM 인스턴스를 업데이트합니다

이제 ASM 인스턴스를 업데이트하여 위치 변경을 반영해야 합니다. 를 클릭합니다 spset 및 pwset 명령은 ASM 디스크 그룹을 시작하는 데 필요한 ASM 메타데이터를 업데이트합니다.

```
ASMCMD> spset +NEWDATA/spfile.ora
ASMCMD> pwset --asm +NEWDATA/orapwasm
```

업데이트된 파일을 사용하여 **ASM**을 활성화합니다

이때 ASM 인스턴스는 이러한 파일의 이전 위치를 계속 사용합니다. 새 위치에서 파일을 다시 읽고 이전 파일의 잠금을 해제하려면 인스턴스를 다시 시작해야 합니다.

```
-bash-4.1$ sqlplus / as sysasm
SQL> shutdown immediate;
ASM diskgroups volume disabled
ASM diskgroups dismounted
ASM instance shutdown
```

```
SQL> startup
ASM instance started
Total System Global Area 1140850688 bytes
Fixed Size 2933400 bytes
Variable Size 1112751464 bytes
ASM Cache 25165824 bytes
ORA-15032: not all alterations performed
ORA-15017: diskgroup "NEWDATA" cannot be mounted
ORA-15013: diskgroup "NEWDATA" is already mounted
```

이전 **spfile** 및 암호 파일을 제거합니다

절차가 성공적으로 수행되면 이전 파일이 더 이상 잠기지 않고 제거할 수 있습니다.

```
-bash-4.1$ asmcmd
ASMCMDB> rm +DATA/spfile.ora
ASMCMDB> rm +DATA/orapwasm
```

Oracle ASM에서 ASM으로의 복사

Oracle ASM은 기본적으로 경량의 복합 볼륨 관리자와 파일 시스템입니다. 파일 시스템이 바로 보이지 않으므로 RMAN을 사용하여 복사 작업을 수행해야 합니다. 복사 기반의 마이그레이션 프로세스는 안전하고 간단하지만 약간의 운영 중단이 발생합니다. 운영 중단을 최소화할 수 있지만 완전히 제거되지는 않습니다.

ASM 기반 데이터베이스의 무중단 마이그레이션을 원하는 경우 ASM의 기능을 활용하여 이전 LUN을 삭제하는 동시에 ASM 익스텐트를 새 LUN으로 재조정하는 것이 가장 좋습니다. 일반적으로 안전하면서 운영 중단이 일어나지 않지만 백아웃 경로는 제공되지 않습니다. 기능 또는 성능 문제가 발생할 경우 데이터를 소스로 다시 마이그레이션하는 방법만 사용할 수 있습니다.

데이터를 이동하지 않고 데이터베이스를 새 위치로 복사하여 원본 데이터를 그대로 유지하면 이러한 위험을 방지할 수 있습니다. 이 데이터베이스는 새 위치에서 완전히 테스트된 후 가동할 수 있으며, 문제가 발견될 경우 원래 데이터베이스를 폴백 옵션으로 사용할 수 있습니다.

이 절차는 RMAN과 관련된 여러 옵션 중 하나입니다. 초기 백업이 생성된 후 나중에 로그 재생을 통해 동기화되는 2단계 프로세스를 허용하도록 설계되었습니다. 이 프로세스는 초기 베이스라인 복사 중에 데이터베이스가 운영 상태를 유지하고 데이터를 제공할 수 있기 때문에 다운타임을 최소화하는 것이 좋습니다.

데이터베이스를 복사합니다

Oracle RMAN은 현재 ASM 디스크 그룹에 있는 소스 데이터베이스의 레벨 0(전체) 복제본을 생성합니다 +DATA 의 새 위치로 이동합니다 +NEWDATA.

```

-bash-4.1$ rman target /
Recovery Manager: Release 12.1.0.2.0 - Production on Sun Dec 6 17:40:03
2015
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights
reserved.
connected to target database: TOAST (DBID=2084313411)
RMAN> backup as copy incremental level 0 database format '+NEWDATA' tag
'ONTAP_MIGRATION';
Starting backup at 06-DEC-15
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=302 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00001
name=+DATA/TOAST/DATAFILE/system.262.897683141
...
input datafile file number=00004
name=+DATA/TOAST/DATAFILE/users.264.897683151
output file name=+NEWDATA/TOAST/DATAFILE/users.258.897759623
tag=ONTAP_MIGRATION RECID=5 STAMP=897759622
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 0 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWDATA/TOAST/BACKUPSET/2015_12_06/nnsnn0_ontap_migration_0.262.89
7759623 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15

```

아카이브 로그 스위치를 강제 적용합니다

아카이브 로그에 복사의 일관성을 완전히 유지하는 데 필요한 모든 데이터가 포함되도록 하려면 아카이브 로그 스위치를 강제로 사용해야 합니다. 이 명령을 사용하지 않으면 주요 데이터가 재실행 로그에 계속 존재할 수 있습니다.

```

RMAN> sql 'alter system archive log current';
sql statement: alter system archive log current

```

원본 데이터베이스를 종료합니다

데이터베이스가 종료되고 제한된 읽기 전용 모드로 전환되기 때문에 이 단계에서 종단이 시작됩니다. 소스 데이터베이스를 종료하려면 다음 명령을 실행합니다.

```

RMAN> shutdown immediate;
using target database control file instead of recovery catalog
database closed
database dismounted
Oracle instance shut down
RMAN> startup mount;
connected to target database (not started)
Oracle instance started
database mounted
Total System Global Area      805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                 390073456 bytes
Database Buffers              406847488 bytes
Redo Buffers                   5455872 bytes

```

제어 파일 백업

마이그레이션을 중단하고 원래 스토리지 위치로 되돌려야 하는 경우 controlfile을 백업해야 합니다. 백업 제어 파일 사본이 100% 필요한 것은 아니지만 데이터베이스 파일 위치를 원래 위치로 다시 설정하는 프로세스가 더 쉬워집니다.

```

RMAN> backup as copy current controlfile format '/tmp/TOAST.ctrl';
Starting backup at 06-DEC-15
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=358 device type=DISK
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/tmp/TOAST.ctrl tag=TAG20151206T174753 RECID=6
STAMP=897760073
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15

```

매개 변수 업데이트

현재 spfile에는 이전 ASM 디스크 그룹 내의 현재 위치에 있는 컨트롤 파일에 대한 참조가 포함되어 있습니다. 중간 pfile 버전을 편집하여 쉽게 편집할 수 있도록 편집해야 합니다.

```

RMAN> create pfile='/tmp/pfile' from spfile;
Statement processed

```

pfile을 업데이트합니다

새 ASM 디스크 그룹 이름을 반영하도록 이전 ASM 디스크 그룹을 참조하는 모든 매개 변수를 업데이트합니다. 그런 다음 업데이트된 pfile을 저장합니다. 를 확인합니다 db_create 매개 변수가 있습니다.

아래 예에서는 예에 대한 참조를 나타냅니다 +DATA 이(가) 로 변경되었습니다 +NEWDATA 노란색으로 강조 표시됩니다. 두 가지 주요 매개 변수는입니다 db_create 올바른 위치에 새 파일을 만드는 매개 변수입니다.

```
*.compatible='12.1.0.2.0'  
*.control_files='+NEWLOGS/TOAST/CONTROLFILE/current.258.897683139'  
*.db_block_size=8192  
*. db_create_file_dest='+NEWDATA'  
*. db_create_online_log_dest_1='+NEWLOGS'  
*.db_domain=''  
*.db_name='TOAST'  
*.diagnostic_dest='/orabin'  
*.dispatchers='(PROTOCOL=TCP) (SERVICE=TOASTXDB) '  
*.log_archive_dest_1='LOCATION='+NEWLOGS'  
*.log_archive_format='%t_%s_%r.dbf'
```

init.ora 파일을 업데이트합니다

대부분의 ASM 기반 데이터베이스는 를 사용합니다 init.ora 에 있는 파일 \$ORACLE_HOME/dbs 디렉토리로, ASM 디스크 그룹의 spfile을 가리킵니다. 이 파일은 새 ASM 디스크 그룹의 위치로 리디렉션되어야 합니다.

```
-bash-4.1$ cd $ORACLE_HOME/dbs  
-bash-4.1$ cat initTOAST.ora  
SPFILE='+DATA/TOAST/spfileTOAST.ora'
```

이 파일을 다음과 같이 변경합니다.

```
SPFILE='+NEWLOGS/TOAST/spfileTOAST.ora
```

매개 변수 파일 재생성

이제 편집된 pfile의 데이터로 spfile을 채울 준비가 되었습니다.

```
RMAN> create spfile from pfile='/tmp/pfile';  
Statement processed
```

데이터베이스를 시작하여 새 **spfile** 사용을 시작합니다

데이터베이스를 시작하여 새로 생성된 spfile을 사용하고 시스템 매개변수에 대한 추가 변경 사항이 올바르게 기록되었는지 확인합니다.

```

RMAN> startup nomount;
connected to target database (not started)
Oracle instance started
Total System Global Area      805306368 bytes
Fixed Size                    2929552 bytes
Variable Size                 373296240 bytes
Database Buffers              423624704 bytes
Redo Buffers                   5455872 bytes

```

제어파일을 복원합니다

RMAN에서 생성된 백업 제어 파일은 RMAN에서 새 spfile에 지정된 위치로 직접 복구할 수도 있습니다.

```

RMAN> restore controlfile from
'+DATA/TOAST/CONTROLFILE/current.258.897683139';
Starting restore at 06-DEC-15
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=417 device type=DISK
channel ORA_DISK_1: copied control file copy
output file name=+NEWLOGS/TOAST/CONTROLFILE/current.273.897761061
Finished restore at 06-DEC-15

```

데이터베이스를 마운트하고 새 컨트롤 파일의 사용을 확인합니다.

```

RMAN> alter database mount;
using target database control file instead of recovery catalog
Statement processed

```

```

SQL> show parameter control_files;
NAME                                TYPE                                VALUE
-----
control_files                       string
+NEWLOGS/TOAST/CONTROLFILE/cur
rent.273.897761061

```

로그 재생

데이터베이스는 현재 이전 위치에 있는 데이터 파일을 사용합니다. 복사본을 사용하려면 먼저 복사본을 동기화해야 합니다. 초기 복제 프로세스 중에 시간이 경과했으며 변경 사항이 주로 아카이브 로그에 기록되었습니다. 이러한 변경 사항은 다음과 같이 복제됩니다.

1. 아카이브 로그가 포함된 RMAN 증분 백업을 수행합니다.

```
RMAN> backup incremental level 1 format '+NEWLOGS' for recover of copy
with tag 'ONTAP_MIGRATION' database;
Starting backup at 06-DEC-15
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=62 device type=DISK
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/TOAST/DATAFILE/system.262.897683141
input datafile file number=00002
name=+DATA/TOAST/DATAFILE/sysaux.260.897683143
input datafile file number=00003
name=+DATA/TOAST/DATAFILE/undotbs1.257.897683145
input datafile file number=00004
name=+DATA/TOAST/DATAFILE/users.264.897683151
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.
897762693 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/ncsnn1_ontap_migration_0.267.
897762697 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15
```

2. 로그를 재생합니다.

```

RMAN> recover copy of database with tag 'ONTAP_MIGRATION';
Starting recover at 06-DEC-15
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00001
name=+NEWDATA/TOAST/DATAFILE/system.259.897759609
recovering datafile copy file number=00002
name=+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615
recovering datafile copy file number=00003
name=+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619
recovering datafile copy file number=00004
name=+NEWDATA/TOAST/DATAFILE/users.258.897759623
channel ORA_DISK_1: reading from backup piece
+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.8977626
93
channel ORA_DISK_1: piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.
897762693 tag=ONTAP_MIGRATION
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished recover at 06-DEC-15

```

활성화

복원된 컨트롤 파일은 원래 위치에 있는 데이터 파일을 참조하며 복사된 데이터 파일의 경로 정보도 포함합니다.

1. 활성 데이터 파일을 변경하려면 `switch database to copy` 명령.

```

RMAN> switch database to copy;
datafile 1 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/system.259.897759609"
datafile 2 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615"
datafile 3 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619"
datafile 4 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/users.258.897759623"

```

활성 데이터 파일은 이제 복사된 데이터 파일이지만 최종 redo 로그에 변경 내용이 포함될 수 있습니다.

2. 나머지 로그를 모두 재생하려면 `recover database` 명령. 메시지가 표시되는 경우 `media recovery complete` 프로세스가 성공했다는 메시지가 나타납니다.

```

RMAN> recover database;
Starting recover at 06-DEC-15
using channel ORA_DISK_1
starting media recovery
media recovery complete, elapsed time: 00:00:01
Finished recover at 06-DEC-15

```

이 프로세스는 일반 데이터 파일의 위치만 변경했습니다. 임시 데이터 파일은 이름을 바꿔야 하지만 임시 파일이므로 복사할 필요가 없습니다. 데이터베이스가 현재 다운되어 임시 데이터 파일에 활성 데이터가 없습니다.

3. 임시 데이터 파일을 재배치하려면 먼저 해당 위치를 확인합니다.

```

RMAN> select file#||' '||name from v$tempfile;
FILE#||' '||NAME
-----
-----
1 +DATA/TOAST/TEMPFILE/temp.263.897683145

```

4. 각 데이터 파일의 새 이름을 설정하는 RMAN 명령을 사용하여 임시 데이터 파일을 재배치합니다. OMF(Oracle Managed Files)에서는 전체 이름이 필요하지 않으며 ASM 디스크 그룹이면 충분합니다. 데이터베이스가 열리면 OMF는 ASM 디스크 그룹의 적절한 위치에 연결됩니다. 파일을 재배치하려면 다음 명령을 실행합니다.

```

run {
set newname for tempfile 1 to '+NEWDATA';
switch tempfile all;
}

```

```

RMAN> run {
2> set newname for tempfile 1 to '+NEWDATA';
3> switch tempfile all;
4> }
executing command: SET NEWNAME
renamed tempfile 1 to +NEWDATA in control file

```

로그 마이그레이션을 다시 실행합니다

마이그레이션 프로세스는 거의 완료되었지만 재실행 로그는 여전히 원본 ASM 디스크 그룹에 있습니다. REDO 로그는 직접 재배치할 수 없습니다. 대신 새 redo 로그 세트가 생성되어 구성에 추가된 다음 이전 로그가 삭제됩니다.

1. 재실행 로그 그룹의 수와 해당 그룹 번호를 식별합니다.

```

RMAN> select group#||' '||member from v$logfile;
GROUP#||' '||MEMBER
-----
-----
1  +DATA/TOAST/ONLINELOG/group_1.261.897683139
2  +DATA/TOAST/ONLINELOG/group_2.259.897683139
3  +DATA/TOAST/ONLINELOG/group_3.256.897683139

```

2. redo 로그의 크기를 입력합니다.

```

RMAN> select group#||' '||bytes from v$log;
GROUP#||' '||BYTES
-----
-----
1  52428800
2  52428800
3  52428800

```

3. 각 redo 로그에 대해 일치하는 구성을 가진 새 그룹을 생성합니다. OMF를 사용하지 않는 경우 전체 경로를 지정해야 합니다. 이는 를 사용하는 예이기도 합니다 db_create_online_log 매개 변수. 앞에서 설명한 것처럼 이 매개 변수는 +NEWLOGS 로 설정되었습니다. 이 구성을 사용하면 파일 위치나 특정 ASM 디스크 그룹을 지정할 필요 없이 다음 명령을 사용하여 새로운 온라인 로그를 생성할 수 있습니다.

```

RMAN> alter database add logfile size 52428800;
Statement processed
RMAN> alter database add logfile size 52428800;
Statement processed
RMAN> alter database add logfile size 52428800;
Statement processed

```

4. 데이터베이스를 엽니다.

```

SQL> alter database open;
Database altered.

```

5. 이전 로그를 삭제합니다.

```

RMAN> alter database drop logfile group 1;
Statement processed

```

6. 활성 로그를 삭제할 수 없는 오류가 발생하면 다음 로그로 스위치를 강제로 전환하여 잠금을 해제하고 글로벌

체크포인트를 강제로 설정합니다. 예를 들면 다음과 같습니다. 이 로그 파일에 활성 데이터가 있기 때문에 이전 위치에 있던 로그 파일 그룹 3을 삭제하려는 시도가 거부되었습니다. 체크포인트 다음에 로그 아카이빙을 수행하면 로그 파일을 삭제할 수 있습니다.

```
RMAN> alter database drop logfile group 3;
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of sql statement command at 12/08/2015 20:23:51
ORA-01623: log 3 is current log for instance TOAST (thread 4) - cannot
drop
ORA-00312: online log 3 thread 1:
'+LOGS/TOAST/ONLINELOG/group_3.259.897563549'
RMAN> alter system switch logfile;
Statement processed
RMAN> alter system checkpoint;
Statement processed
RMAN> alter database drop logfile group 3;
Statement processed
```

7. 환경을 검토하여 모든 위치 기반 매개 변수가 업데이트되었는지 확인합니다.

```
SQL> select name from v$datafile;
SQL> select member from v$logfile;
SQL> select name from v$tempfile;
SQL> show parameter spfile;
SQL> select name, value from v$parameter where value is not null;
```

8. 다음 스크립트는 이 프로세스를 단순화하는 방법을 보여 줍니다.

```

[root@host1 current]# ./checkdbdata.pl TOAST
TOAST datafiles:
+NEWDATA/TOAST/DATAFILE/system.259.897759609
+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615
+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619
+NEWDATA/TOAST/DATAFILE/users.258.897759623
TOAST redo logs:
+NEWLOGS/TOAST/ONLINELOG/group_4.266.897763123
+NEWLOGS/TOAST/ONLINELOG/group_5.265.897763125
+NEWLOGS/TOAST/ONLINELOG/group_6.264.897763125
TOAST temp datafiles:
+NEWDATA/TOAST/TEMPFILE/temp.260.897763165
TOAST spfile
spfile                                string
+NEWDATA/spfiletoast.ora
TOAST key parameters
control_files +NEWLOGS/TOAST/CONTROLFILE/current.273.897761061
log_archive_dest_1 LOCATION=+NEWLOGS
db_create_file_dest +NEWDATA
db_create_online_log_dest_1 +NEWLOGS

```

9. ASM 디스크 그룹이 완전히 비워진 경우 에서 디스크 그룹을 마운트 해제할 수 있습니다 `asmcmd`. 그러나 대부분의 경우 다른 데이터베이스 또는 ASM `spfile/passwd` 파일에 속하는 파일이 여전히 존재할 수 있습니다.

```

-bash-4.1$ . oraenv
ORACLE_SID = [TOAST] ? +ASM
The Oracle base remains unchanged with value /orabin
-bash-4.1$ asmcmd
ASMCMD> umount DATA
ASMCMD>

```

Oracle ASM에서 파일 시스템 복사입니다

Oracle ASM-파일 시스템 복사 절차는 ASM과 ASM/ASM 복제 절차와 매우 유사하며, 이점과 제한 사항이 유사합니다. 기본적인 차이점은 ASM 디스크 그룹과 달리 가시적인 파일 시스템을 사용할 때 다양한 명령 및 구성 매개 변수의 구문입니다.

데이터베이스를 복사합니다

Oracle RMAN은 현재 ASM 디스크 그룹에 위치한 소스 데이터베이스의 레벨 0(전체) 복제본을 생성하는 데 사용됩니다 +DATA 의 새 위치로 이동합니다 /oradata.

```

RMAN> backup as copy incremental level 0 database format
'/oradata/TOAST/%U' tag 'ONTAP_MIGRATION';
Starting backup at 13-MAY-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=377 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00001 name=+ASM0/TOAST/system01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-
1_01r5fhjg tag=ONTAP_MIGRATION RECID=1 STAMP=911722099
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
input datafile file number=00002 name=+ASM0/TOAST/sysaux01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-
2_02r5fhjo tag=ONTAP_MIGRATION RECID=2 STAMP=911722106
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
input datafile file number=00003 name=+ASM0/TOAST/undotbs101.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-
3_03r5fhjt tag=ONTAP_MIGRATION RECID=3 STAMP=911722113
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/oradata/TOAST/cf_D-TOAST_id-2098173325_04r5fhk5
tag=ONTAP_MIGRATION RECID=4 STAMP=911722118
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00004 name=+ASM0/TOAST/users01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-
4_05r5fhk6 tag=ONTAP_MIGRATION RECID=5 STAMP=911722118
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 0 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 13-MAY-16
channel ORA_DISK_1: finished piece 1 at 13-MAY-16
piece handle=/oradata/TOAST/06r5fhk7_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-MAY-16

```

아카이브 로그 스위치를 강제 적용합니다

아카이브 로그 스위치를 강제로 사용하면 아카이브 로그에 복제본의 일관성을 완전히 유지하는 데 필요한 모든 데이터가 포함되도록 할 수 있습니다. 이 명령을 사용하지 않으면 주요 데이터가 재실행 로그에 계속 존재할 수 있습니다. 아카이브 로그 스위치를 강제로 전환하려면 다음 명령을 실행합니다.

```
RMAN> sql 'alter system archive log current';
sql statement: alter system archive log current
```

원본 데이터베이스를 종료합니다

데이터베이스가 종료되고 제한된 액세스 읽기 전용 모드로 전환되기 때문에 이 단계에서 중단이 시작됩니다. 소스 데이터베이스를 종료하려면 다음 명령을 실행합니다.

```
RMAN> shutdown immediate;
using target database control file instead of recovery catalog
database closed
database dismounted
Oracle instance shut down
RMAN> startup mount;
connected to target database (not started)
Oracle instance started
database mounted
Total System Global Area      805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                  331353200 bytes
Database Buffers               465567744 bytes
Redo Buffers                    5455872 bytes
```

제어 파일 백업

마이그레이션을 중단하고 원래 스토리지 위치로 되돌려야 하는 경우 제어 파일을 백업합니다. 백업 제어 파일 사본이 100% 필요한 것은 아니지만 데이터베이스 파일 위치를 원래 위치로 다시 설정하는 프로세스가 더 쉬워집니다.

```
RMAN> backup as copy current controlfile format '/tmp/TOAST.ctrl';
Starting backup at 08-DEC-15
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/tmp/TOAST.ctrl tag=TAG20151208T194540 RECID=30
STAMP=897939940
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 08-DEC-15
```

매개 변수 업데이트

```
RMAN> create pfile='/tmp/pfile' from spfile;
Statement processed
```

pfile을 업데이트합니다

이전 ASM 디스크 그룹을 참조하는 모든 매개 변수는 업데이트되어야 하며, 경우에 따라 더 이상 관련이 없을 때 삭제해야 합니다. 새 파일 시스템 경로를 반영하도록 이 경로를 업데이트하고 업데이트된 pfile을 저장합니다. 전체 대상 경로가 나열되어 있는지 확인합니다. 이러한 매개 변수를 업데이트하려면 다음 명령을 실행합니다.

```
*.audit_file_dest='/orabin/admin/TOAST/adump'  
*.audit_trail='db'  
*.compatible='12.1.0.2.0'  
*.control_files='/logs/TOAST/arch/control01.ctl','/logs/TOAST/redo/control  
02.ctl'  
*.db_block_size=8192  
*.db_domain=''  
*.db_name='TOAST'  
*.diagnostic_dest='/orabin'  
*.dispatchers='(PROTOCOL=TCP) (SERVICE=TOASTXDB)'  
*.log_archive_dest_1='LOCATION=/logs/TOAST/arch'  
*.log_archive_format='%t_%s_%r.dbf'  
*.open_cursors=300  
*.pga_aggregate_target=256m  
*.processes=300  
*.remote_login_passwordfile='EXCLUSIVE'  
*.sga_target=768m  
*.undo_tablespace='UNDOTBS1'
```

원본 init.ora 파일을 비활성화합니다

이 파일은 에 있습니다 \$ORACLE_HOME/dbs 디렉토리이며 일반적으로 ASM 디스크 그룹의 spfile에 대한 포인터로 사용되는 pfile에 있습니다. 원본 spfile이 더 이상 사용되지 않도록 하려면 이름을 바꿉니다. 그러나 마이그레이션을 중단해야 하는 경우 이 파일이 필요하므로 삭제하지 마십시오.

```
[oracle@jfsc1 ~]$ cd $ORACLE_HOME/dbs  
[oracle@jfsc1 dbs]$ cat initTOAST.ora  
SPFILE='+ASM0/TOAST/spfileTOAST.ora'  
[oracle@jfsc1 dbs]$ mv initTOAST.ora initTOAST.ora.prev  
[oracle@jfsc1 dbs]$
```

매개 변수 파일 재생성

이 단계는 spfile 재배포의 마지막 단계입니다. 원본 spfile은 더 이상 사용되지 않으며 현재 중간 파일을 사용하여 데이터베이스가 시작(마운트되지 않음)됩니다. 이 파일의 내용은 다음과 같이 새 spfile 위치에 쓸 수 있습니다.

```
RMAN> create spfile from pfile='/tmp/pfile';  
Statement processed
```

데이터베이스를 시작하여 새 **spfile** 사용을 시작합니다

중간 파일의 잠금을 해제하고 새 spfile 파일만 사용하여 데이터베이스를 시작하려면 데이터베이스를 시작해야 합니다. 데이터베이스를 시작하면 새 spfile 위치가 올바르고 데이터가 유효하다는 것도 증명됩니다.

```
RMAN> shutdown immediate;
Oracle instance shut down
RMAN> startup nomount;
connected to target database (not started)
Oracle instance started
Total System Global Area      805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                 331353200 bytes
Database Buffers              465567744 bytes
Redo Buffers                   5455872 bytes
```

제어파일을 복원합니다

경로에 백업 제어 파일이 생성되었습니다 /tmp/TOAST.ctrl 절차의 앞부분에 있습니다. 새 spfile은 제어 파일 위치를 로 정의합니다 /logfs/TOAST/ctrl/ctrlfile1.ctrl 및 /logfs/TOAST/redo/ctrlfile2.ctrl. 그러나 해당 파일은 아직 존재하지 않습니다.

1. 이 명령은 컨트롤 파일 데이터를 spfile에 정의된 경로로 복원합니다.

```
RMAN> restore controlfile from '/tmp/TOAST.ctrl';
Starting restore at 13-MAY-16
using channel ORA_DISK_1
channel ORA_DISK_1: copied control file copy
output file name=/logs/TOAST/arch/control01.ctrl
output file name=/logs/TOAST/redo/control02.ctrl
Finished restore at 13-MAY-16
```

2. mount 명령을 실행하여 제어 파일이 올바르게 검색되고 유효한 데이터가 포함되도록 합니다.

```
RMAN> alter database mount;
Statement processed
released channel: ORA_DISK_1
```

를 확인합니다 control_files 매개 변수에서 다음 명령을 실행합니다.

```
SQL> show parameter control_files;
NAME                                TYPE                                VALUE
-----                                -
control_files                        string
/logs/TOAST/arch/control01.ctl
,
/logs/TOAST/redo/control02.c
tl
```

로그 재생

데이터베이스가 현재 이전 위치의 데이터 파일을 사용하고 있습니다. 복사본을 사용하려면 먼저 데이터 파일을 동기화해야 합니다. 초기 복제 프로세스 중에 시간이 경과했으며 변경 사항은 주로 아카이브 로그에 기록되었습니다. 이러한 변경 사항은 다음 두 단계에 복제됩니다.

1. 아카이브 로그가 포함된 RMAN 증분 백업을 수행합니다.

```
RMAN> backup incremental level 1 format '/logs/TOAST/arch/%U' for
recover of copy with tag 'ONTAP_MIGRATION' database;
Starting backup at 13-MAY-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=124 device type=DISK
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=+ASM0/TOAST/system01.dbf
input datafile file number=00002 name=+ASM0/TOAST/sysaux01.dbf
input datafile file number=00003 name=+ASM0/TOAST/undotbs101.dbf
input datafile file number=00004 name=+ASM0/TOAST/users01.dbf
channel ORA_DISK_1: starting piece 1 at 13-MAY-16
channel ORA_DISK_1: finished piece 1 at 13-MAY-16
piece handle=/logs/TOAST/arch/09r5fj8i_1_1 tag=ONTAP_MIGRATION
comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-MAY-16
RMAN-06497: WARNING: control file is not current, control file
AUTOBACKUP skipped
```

2. 로그를 재생합니다.

```

RMAN> recover copy of database with tag 'ONTAP_MIGRATION';
Starting recover at 13-MAY-16
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00001 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
recovering datafile copy file number=00002 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
recovering datafile copy file number=00003 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt
recovering datafile copy file number=00004 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
channel ORA_DISK_1: reading from backup piece
/logs/TOAST/arch/09r5fj8i_1_1
channel ORA_DISK_1: piece handle=/logs/TOAST/arch/09r5fj8i_1_1
tag=ONTAP_MIGRATION
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished recover at 13-MAY-16
RMAN-06497: WARNING: control file is not current, control file
AUTOBACKUP skipped

```

활성화

복원된 컨트롤 파일은 원래 위치에 있는 데이터 파일을 참조하며 복사된 데이터 파일의 경로 정보도 포함합니다.

1. 활성 데이터 파일을 변경하려면 `switch database to copy` 명령:

```

RMAN> switch database to copy;
datafile 1 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-SYSTEM_FNO-1_01r5fhjg"
datafile 2 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-SYSAUX_FNO-2_02r5fhjo"
datafile 3 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt"
datafile 4 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-USERS_FNO-4_05r5fhk6"

```

2. 데이터 파일은 완전히 일관되어야 하지만 온라인 재실행 로그에 기록된 나머지 변경 내용을 재생하려면 마지막 단계가 필요합니다. `recover database` 명령을 사용하여 이러한 변경 사항을 재생하고 복사본을 원본과 100% 동일하게 만듭니다. 하지만 복사본이 아직 열려 있지 않습니다.

```

RMAN> recover database;
Starting recover at 13-MAY-16
using channel ORA_DISK_1
starting media recovery
archived log for thread 1 with sequence 28 is already on disk as file
+ASM0/TOAST/redo01.log
archived log file name=+ASM0/TOAST/redo01.log thread=1 sequence=28
media recovery complete, elapsed time: 00:00:00
Finished recover at 13-MAY-16

```

임시 데이터 파일 재배치

1. 원본 디스크 그룹에서 여전히 사용 중인 임시 데이터 파일의 위치를 식별합니다.

```

RMAN> select file#||' '||name from v$tempfile;
FILE#||' '||NAME
-----
-----
1 +ASM0/TOAST/temp01.dbf

```

2. 데이터 파일을 재배치하려면 다음 명령을 실행합니다. tempfiles가 많은 경우 텍스트 편집기를 사용하여 RMAN 명령을 생성한 다음 잘라내어 붙여 넣습니다.

```

RMAN> run {
2> set newname for tempfile 1 to '/oradata/TOAST/temp01.dbf';
3> switch tempfile all;
4> }
executing command: SET NEWNAME
renamed tempfile 1 to /oradata/TOAST/temp01.dbf in control file

```

로그 마이그레이션을 다시 실행합니다

마이그레이션 프로세스는 거의 완료되었지만 재실행 로그는 여전히 원본 ASM 디스크 그룹에 있습니다. REDO 로그는 직접 재배치할 수 없습니다. 대신 새 redo 로그 세트가 생성되고 구성에 추가되며, 그 다음에 이전 로그가 삭제됩니다.

1. 재실행 로그 그룹의 수와 해당 그룹 번호를 식별합니다.

```

RMAN> select group#||' '||member from v$logfile;
GROUP#||' '||MEMBER
-----
-----
1 +ASM0/TOAST/redo01.log
2 +ASM0/TOAST/redo02.log
3 +ASM0/TOAST/redo03.log

```

2. redo 로그의 크기를 입력합니다.

```

RMAN> select group#||' '||bytes from v$log;
GROUP#||' '||BYTES
-----
-----
1 52428800
2 52428800
3 52428800

```

3. 각 redo 로그에 대해 새 파일 시스템 위치를 사용하여 현재 redo 로그 그룹과 동일한 크기를 사용하여 새 그룹을 생성합니다.

```

RMAN> alter database add logfile '/logs/TOAST/redo/log00.rdo' size
52428800;
Statement processed
RMAN> alter database add logfile '/logs/TOAST/redo/log01.rdo' size
52428800;
Statement processed
RMAN> alter database add logfile '/logs/TOAST/redo/log02.rdo' size
52428800;
Statement processed

```

4. 이전 스토리지에 있는 이전 로그 파일 그룹을 제거합니다.

```

RMAN> alter database drop logfile group 4;
Statement processed
RMAN> alter database drop logfile group 5;
Statement processed
RMAN> alter database drop logfile group 6;
Statement processed

```

5. 활성 로그를 삭제하는 블록에 오류가 발생하는 경우 다음 로그로 스위치를 강제 전환하여 잠금을 해제하고 글로벌 체크포인트를 강제 적용합니다. 예를 들면 다음과 같습니다. 이 로그 파일에 활성 데이터가 있기 때문에 이전 위치에

있던 로그 파일 그룹 3을 삭제하려는 시도가 거부되었습니다. 로그 아카이빙과 체크포인트가 지나면 로그 파일을 삭제할 수 있습니다.

```

RMAN> alter database drop logfile group 4;
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of sql statement command at 12/08/2015 20:23:51
ORA-01623: log 4 is current log for instance TOAST (thread 4) - cannot
drop
ORA-00312: online log 4 thread 1:
'+NEWLOGS/TOAST/ONLINELOG/group_4.266.897763123'
RMAN> alter system switch logfile;
Statement processed
RMAN> alter system checkpoint;
Statement processed
RMAN> alter database drop logfile group 4;
Statement processed

```

6. 환경을 검토하여 모든 위치 기반 매개 변수가 업데이트되었는지 확인합니다.

```

SQL> select name from v$datafile;
SQL> select member from v$logfile;
SQL> select name from v$tempfile;
SQL> show parameter spfile;
SQL> select name, value from v$parameter where value is not null;

```

7. 다음 스크립트는 이 프로세스를 보다 쉽게 만드는 방법을 보여 줍니다.

```

[root@jfscl current]# ./checkdbdata.pl TOAST
TOAST datafiles:
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
TOAST redo logs:
/logs/TOAST/redo/log00.rdo
/logs/TOAST/redo/log01.rdo
/logs/TOAST/redo/log02.rdo
TOAST temp datafiles:
/oradata/TOAST/temp01.dbf
TOAST spfile
spfile                                string
/orabin/product/12.1.0/dbhome_
                                         1/dbs/spfileTOAST.ora
TOAST key parameters
control_files /logs/TOAST/arch/control01.ctl,
/logs/TOAST/redo/control02.ctl
log_archive_dest_1 LOCATION=/logs/TOAST/arch

```

8. ASM 디스크 그룹이 완전히 비워진 경우 에서 디스크 그룹을 마운트 해제할 수 있습니다 `asmcmd`. 대부분의 경우 다른 데이터베이스 또는 ASM `spfile/passwd` 파일에 속하는 파일이 계속 존재할 수 있습니다.

```

-bash-4.1$ . oraenv
ORACLE_SID = [TOAST] ? +ASM
The Oracle base remains unchanged with value /orabin
-bash-4.1$ asmcmd
ASMCMD> umount DATA
ASMCMD>

```

데이터 파일 정리 절차

마이그레이션 프로세스로 인해 Oracle RMAN의 사용 방식에 따라 긴 구문 또는 암호화된 데이터 파일이 생성될 수 있습니다. 여기에 표시된 예에서는 의 파일 형식으로 백업이 수행되었습니다 `/oradata/TOAST/%U.%U RMAN`이 각 데이터 파일에 대해 기본 고유 이름을 생성해야 함을 나타냅니다. 결과는 다음 텍스트에 표시된 것과 유사합니다. 데이터 파일의 기존 이름은 이름 안에 포함됩니다. 이 작업은 에 나와 있는 스크립트된 접근 방식을 사용하여 정리할 수 있습니다 "[ASM 마이그레이션 정리](#)".

```
[root@jfscl current]# ./fixuniquenames.pl TOAST
#sqlplus Commands
shutdown immediate;
startup mount;
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
/oradata/TOAST/system.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
/oradata/TOAST/sysaux.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-
3_03r5fhjt /oradata/TOAST/undotbs1.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
/oradata/TOAST/users.dbf
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
SYSTEM_FNO-1_01r5fhjg' to '/oradata/TOAST/system.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
SYSAUX_FNO-2_02r5fhjo' to '/oradata/TOAST/sysaux.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
UNDOTBS1_FNO-3_03r5fhjt' to '/oradata/TOAST/undotbs1.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
USERS_FNO-4_05r5fhk6' to '/oradata/TOAST/users.dbf';
alter database open;
```

Oracle ASM 재조정

앞서 설명한 대로 재조정 프로세스를 사용하여 Oracle ASM 디스크 그룹을 새 스토리지 시스템으로 투명하게 마이그레이션할 수 있습니다. 요약하면 재조정 프로세스에서는 크기가 동일한 LUN을 기존 LUN 그룹에 추가한 다음 이전 LUN의 삭제 작업을 수행해야 합니다. Oracle ASM은 기본 데이터를 최적의 레이아웃으로 새 스토리지로 자동으로 재이동한 다음 완료되면 이전 LUN을 해제합니다.

마이그레이션 프로세스는 효율적인 순차적 I/O를 사용하며 일반적으로 성능 중단을 일으키지 않지만 필요할 때 마이그레이션 속도를 조절할 수 있습니다.

마이그레이션할 데이터를 식별합니다

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||'
' ||header_status from v$asm_disk;
NEWDATA_0003 1 10240 /dev/mapper/3600a098038303537762b47594c315864 MEMBER
NEWDATA_0002 1 10240 /dev/mapper/3600a098038303537762b47594c315863 MEMBER
NEWDATA_0000 1 10240 /dev/mapper/3600a098038303537762b47594c315861 MEMBER
NEWDATA_0001 1 10240 /dev/mapper/3600a098038303537762b47594c315862 MEMBER
SQL> select group_number||' '||name from v$asm_diskgroup;
1 NEWDATA
```

새 LUN을 생성합니다

동일한 크기의 새 LUN을 생성하고 필요에 따라 사용자 및 그룹 멤버십을 설정합니다. LUN은 로 표시되어야 합니다 CANDIDATE 디스크.

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||'
'||header_status from v$asm_disk;
0 0 /dev/mapper/3600a098038303537762b47594c31586b CANDIDATE
0 0 /dev/mapper/3600a098038303537762b47594c315869 CANDIDATE
0 0 /dev/mapper/3600a098038303537762b47594c315858 CANDIDATE
0 0 /dev/mapper/3600a098038303537762b47594c31586a CANDIDATE
NEWDATA_0003 1 10240 /dev/mapper/3600a098038303537762b47594c315864 MEMBER
NEWDATA_0002 1 10240 /dev/mapper/3600a098038303537762b47594c315863 MEMBER
NEWDATA_0000 1 10240 /dev/mapper/3600a098038303537762b47594c315861 MEMBER
NEWDATA_0001 1 10240 /dev/mapper/3600a098038303537762b47594c315862 MEMBER
```

새 LUN을 추가합니다

추가 및 삭제 작업은 함께 수행할 수 있지만 일반적으로 두 단계로 새 LUN을 추가하는 것이 더 쉽습니다. 먼저 새 LUN을 디스크 그룹에 추가합니다. 이 단계를 수행하면 익스텐트의 절반이 현재 ASM LUN에서 새 LUN으로 마이그레이션됩니다.

재조정 성능은 데이터가 전송되는 속도를 나타냅니다. 숫자가 클수록 데이터 전송의 병렬 처리 수가 높아집니다. 마이그레이션은 성능 문제를 일으킬 소지가 없는 효율적인 순차적 I/O 작업을 통해 수행됩니다. 그러나 필요한 경우 진행 중인 마이그레이션의 균형 조정 성능을 로 조정할 수 있습니다 alter diskgroup [name] rebalance power [level] 명령. 일반적인 마이그레이션은 5의 값을 사용합니다.

```
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c31586b' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c315869' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c315858' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c31586a' rebalance power 5;
Diskgroup altered.
```

작동을 모니터링합니다

재조정 작업을 여러 방법으로 모니터링하고 관리할 수 있습니다. 이 예에서는 다음 명령을 사용했습니다.

```
SQL> select group_number,operation,state from v$asm_operation;
GROUP_NUMBER OPERA STAT
-----
1 REBAL RUN
1 REBAL WAIT
```

마이그레이션이 완료되면 재조정 작업이 보고되지 않습니다.

```
SQL> select group_number,operation,state from v$asm_operation;
no rows selected
```

기존 LUN을 삭제합니다

이제 마이그레이션이 절반 정도 완료되었습니다. 몇 가지 기본 성능 테스트를 수행하여 환경이 양호한지 확인하는 것이 좋습니다. 확인 후 이전 LUN을 삭제하여 나머지 데이터를 재배포할 수 있습니다. 그러나 LUN이 즉시 해제되지는 않습니다. 삭제 작업은 Oracle ASM에 먼저 익스텐트를 재배포한 다음 LUN을 해제하라는 신호를 보냅니다.

```
sqlplus / as sysasm
SQL> alter diskgroup NEWDATA drop disk NEWDATA_0000 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA drop disk NEWDATA_0001 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup newdata drop disk NEWDATA_0002 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup newdata drop disk NEWDATA_0003 rebalance power 5;
Diskgroup altered.
```

작동을 모니터링합니다

재조정 작업은 여러 가지 방법으로 모니터링 및 관리할 수 있습니다. 이 예에서는 다음 명령을 사용했습니다.

```
SQL> select group_number,operation,state from v$asm_operation;
GROUP_NUMBER OPERA STAT
-----
1 REBAL RUN
1 REBAL WAIT
```

마이그레이션이 완료되면 재조정 작업이 보고되지 않습니다.

```
SQL> select group_number,operation,state from v$asm_operation;
no rows selected
```

이전 LUN을 제거합니다

디스크 그룹에서 기존 LUN을 제거하기 전에 헤더 상태에 대한 최종 확인 작업을 수행해야 합니다. ASM에서 LUN을 릴리즈하면 더 이상 이름이 나열되지 않고 헤더 상태가 로 표시됩니다 FORMER. 이는 이러한 LUN을 시스템에서 안전하게 제거할 수 있음을 나타냅니다.

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||'
'||header_status from v$asm_disk;
NAME||' '||GROUP_NUMBER||' '||TOTAL_MB||' '||PATH||' '||HEADER_STATUS
-----
-----
0 0 /dev/mapper/3600a098038303537762b47594c315863 FORMER
0 0 /dev/mapper/3600a098038303537762b47594c315864 FORMER
0 0 /dev/mapper/3600a098038303537762b47594c315861 FORMER
0 0 /dev/mapper/3600a098038303537762b47594c315862 FORMER
NEWDATA_0005 1 10240 /dev/mapper/3600a098038303537762b47594c315869 MEMBER
NEWDATA_0007 1 10240 /dev/mapper/3600a098038303537762b47594c31586a MEMBER
NEWDATA_0004 1 10240 /dev/mapper/3600a098038303537762b47594c31586b MEMBER
NEWDATA_0006 1 10240 /dev/mapper/3600a098038303537762b47594c315858 MEMBER
8 rows selected.
```

LVM 마이그레이션

여기에 제시된 절차는 라는 볼륨 그룹의 LVM 기반 마이그레이션 원칙을 보여줍니다 datavg. 그 예가 Linux LVM에서 도출되었지만 원칙은 AIX, HP-UX 및 VxVM에도 동일하게 적용됩니다. 정확한 명령은 다를 수 있습니다.

1. 현재 에 있는 LUN을 식별합니다 datavg 볼륨 그룹:

```
[root@host1 ~]# pvdisplay -C | grep datavg
/dev/mapper/3600a098038303537762b47594c31582f datavg lvm2 a-- 10.00g
10.00g
/dev/mapper/3600a098038303537762b47594c31585a datavg lvm2 a-- 10.00g
10.00g
/dev/mapper/3600a098038303537762b47594c315859 datavg lvm2 a-- 10.00g
10.00g
/dev/mapper/3600a098038303537762b47594c31586c datavg lvm2 a-- 10.00g
10.00g
```

2. 물리적 크기가 같거나 약간 더 큰 새 LUN을 생성하고 물리적 볼륨으로 정의합니다.

```

[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315864
Physical volume "/dev/mapper/3600a098038303537762b47594c315864"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315863
Physical volume "/dev/mapper/3600a098038303537762b47594c315863"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315862
Physical volume "/dev/mapper/3600a098038303537762b47594c315862"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315861
Physical volume "/dev/mapper/3600a098038303537762b47594c315861"
successfully created

```

3. 새 볼륨을 볼륨 그룹에 추가합니다.

```

[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315864
Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315863
Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315862
Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315861
Volume group "datavg" successfully extended

```

4. 를 발행합니다 pvmove 명령을 사용하여 각 현재 LUN의 익스텐트를 새 LUN으로 재배포합니다. 를 클릭합니다 - i [seconds] argument 는 작업의 진행률을 모니터링합니다.

```

[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31582f
/dev/mapper/3600a098038303537762b47594c315864
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 14.2%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 28.4%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 42.5%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 57.1%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 72.3%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 87.3%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31585a
/dev/mapper/3600a098038303537762b47594c315863
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 14.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 29.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 44.8%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 60.1%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 75.8%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 90.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c315859
/dev/mapper/3600a098038303537762b47594c315862
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 14.8%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 29.8%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 45.5%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 61.1%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 76.6%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 91.7%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31586c
/dev/mapper/3600a098038303537762b47594c315861
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 15.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 30.4%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 46.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 61.4%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 77.2%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 92.3%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 100.0%

```

5. 이 프로세스가 완료되면 `lv` 사용하여 볼륨 그룹에서 이전 LUN을 삭제합니다 `vgreduce` 명령. 성공하면 이제 시스템에서 LUN을 안전하게 제거할 수 있습니다.

```
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31582f
Removed "/dev/mapper/3600a098038303537762b47594c31582f" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31585a
Removed "/dev/mapper/3600a098038303537762b47594c31585a" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c315859
Removed "/dev/mapper/3600a098038303537762b47594c315859" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31586c
Removed "/dev/mapper/3600a098038303537762b47594c31586c" from volume
group "datavg"
```

외부 LUN 임포트

FLI-Planning을 사용한 Oracle 마이그레이션

FLI를 사용하여 SAN 리소스를 마이그레이션하는 절차는 NetApp에 설명되어 있습니다 "[TR-4380: 외부 LUN 가져오기를 사용한 SAN 마이그레이션](#)".

데이터베이스 및 호스트 관점에서 볼 때 특별한 단계가 필요하지 않습니다. FC 존이 업데이트되고 LUN을 ONTAP에서 사용할 수 있게 되면 LVM이 LUN에서 LVM 메타데이터를 읽을 수 있어야 합니다. 또한 볼륨 그룹을 추가 구성 단계 없이 사용할 수 있습니다. 드문 경우지만 환경에 이전 스토리지 어레이를 참조하는 하드 코딩된 구성 파일이 포함될 수 있습니다. 예를 들어, 을 포함하는 Linux 시스템이 해당됩니다 `/etc/multipath.conf` 주어진 디바이스의 WWN을 참조하는 규칙은 FLI에 의해 도입된 변경 사항을 반영하도록 업데이트되어야 합니다.



지원되는 구성에 대한 자세한 내용은 NetApp 호환성 매트릭스를 참조하십시오. 사용 환경이 포함되지 않은 경우 NetApp 담당자에게 문의하십시오.

이 예제는 Linux 서버에서 호스팅되는 ASM 및 LVM LUN의 마이그레이션을 보여줍니다. FLI는 다른 운영 체제에서 지원되며 호스트측 명령이 다를 수 있지만 원칙이 동일하고 ONTAP 절차는 동일합니다.

LVM LUN을 식별합니다

준비의 첫 번째 단계는 마이그레이션할 LUN을 식별하는 것입니다. 여기에 나와 있는 예에서는 SAN 기반 파일 시스템 두 개가 에 마운트되어 있습니다 `/orabin` 및 `/backups`.

```
[root@host1 ~]# df -k
Filesystem                1K-blocks      Used Available Use%
Mounted on
/dev/mapper/rhel-root      52403200    8811464  43591736  17% /
devtmpfs                  65882776         0  65882776   0% /dev
...
fas8060-nfs-public:/install 199229440 119368128  79861312  60%
/install
/dev/mapper/sanvg-lvorabin  20961280 12348476   8612804  59%
/orabin
/dev/mapper/sanvg-lvbackups 73364480  62947536 10416944  86%
/backups
```

볼륨 그룹의 이름은 (볼륨 그룹 이름) - (논리적 볼륨 이름) 형식을 사용하는 디바이스 이름에서 추출할 수 있습니다. 이 경우 볼륨 그룹이 호출됩니다 sanvg.

를 클릭합니다 pvdisplay 명령을 사용하여 다음 명령을 사용하여 이 볼륨 그룹을 지원하는 LUN을 식별할 수 있습니다. 이 경우 를 구성하는 LUN이 10개 있습니다 sanvg 볼륨 그룹:

```
[root@host1 ~]# pvdisplay -C -o pv_name,pv_size,pv_fmt,vg_name
PV                               PSize  VG
/dev/mapper/3600a0980383030445424487556574266 10.00g sanvg
/dev/mapper/3600a0980383030445424487556574267 10.00g sanvg
/dev/mapper/3600a0980383030445424487556574268 10.00g sanvg
/dev/mapper/3600a0980383030445424487556574269 10.00g sanvg
/dev/mapper/3600a098038303044542448755657426a 10.00g sanvg
/dev/mapper/3600a098038303044542448755657426b 10.00g sanvg
/dev/mapper/3600a098038303044542448755657426c 10.00g sanvg
/dev/mapper/3600a098038303044542448755657426d 10.00g sanvg
/dev/mapper/3600a098038303044542448755657426e 10.00g sanvg
/dev/mapper/3600a098038303044542448755657426f 10.00g sanvg
/dev/sda2                               278.38g rhel
```

ASM LUN 식별

ASM LUN도 마이그레이션해야 합니다. sysasm 사용자로 sqlplus에서 LUN 및 LUN 경로 수를 가져오려면 다음 명령을 실행합니다.

```
SQL> select path||' '||os_mb from v$asm_disk;
PATH||' '||OS_MB
-----
-----
/dev/oracleasm/disks/ASM0 10240
/dev/oracleasm/disks/ASM9 10240
/dev/oracleasm/disks/ASM8 10240
/dev/oracleasm/disks/ASM7 10240
/dev/oracleasm/disks/ASM6 10240
/dev/oracleasm/disks/ASM5 10240
/dev/oracleasm/disks/ASM4 10240
/dev/oracleasm/disks/ASM1 10240
/dev/oracleasm/disks/ASM3 10240
/dev/oracleasm/disks/ASM2 10240
10 rows selected.
SQL>
```

FC 네트워크 변경 사항

현재 환경에는 마이그레이션할 LUN 20개가 포함되어 있습니다. ONTAP가 현재 LUN에 액세스할 수 있도록 현재 SAN을 업데이트합니다. 데이터가 아직 마이그레이션되지 않았지만 ONTAP는 현재 LUN에서 구성 정보를 읽어 해당 데이터의 새 홈을 생성해야 합니다.

AFF/FAS 시스템에서 하나 이상의 HBA 포트를 이니시에이터 포트에 구성해야 합니다. 또한 ONTAP이 외부 스토리지 어레이의 LUN에 액세스할 수 있도록 FC 존을 업데이트해야 합니다. 일부 스토리지에는 LUN 마스킹이 구성되어 있으며, 이 경우 특정 LUN을 액세스할 수 있는 WWN이 제한됩니다. 이 경우 LUN 마스킹도 업데이트하여 ONTAP WWN에 대한 액세스 권한을 부여해야 합니다.

이 단계가 완료된 후 ONTAP는 를 사용하여 외부 스토리지 어레이를 볼 수 있어야 합니다 `storage array show` 명령. 반환되는 키 필드는 시스템에서 외부 LUN을 식별하는 데 사용되는 접두사입니다. 아래 예에서는 외부 스토리지의 LUN을 보여 줍니다 FOREIGN_1 의 접두사를 사용하여 ONTAP 내에 나타냅니다 FOR-1.

외부 어레이를 식별합니다

```
Cluster01::> storage array show -fields name,prefix
name          prefix
-----
FOREIGN_1     FOR-1
Cluster01::>
```

외부 LUN 식별

LUN은 를 전달하여 나열할 수 있습니다 `array-name` 를 누릅니다 `storage disk show` 명령. 반환되는 데이터는 마이그레이션 절차 중에 여러 번 참조됩니다.

```

Cluster01::> storage disk show -array-name FOREIGN_1 -fields disk,serial
disk      serial-number
-----
FOR-1.1   800DT$HuVWBX
FOR-1.2   800DT$HuVWBZ
FOR-1.3   800DT$HuVWBW
FOR-1.4   800DT$HuVWBV
FOR-1.5   800DT$HuVWB/
FOR-1.6   800DT$HuVWBa
FOR-1.7   800DT$HuVWBd
FOR-1.8   800DT$HuVWBb
FOR-1.9   800DT$HuVWBc
FOR-1.10  800DT$HuVWBe
FOR-1.11  800DT$HuVWBf
FOR-1.12  800DT$HuVWBg
FOR-1.13  800DT$HuVWBh
FOR-1.14  800DT$HuVWBh
FOR-1.15  800DT$HuVWBj
FOR-1.16  800DT$HuVWBk
FOR-1.17  800DT$HuVWBm
FOR-1.18  800DT$HuVWBn
FOR-1.19  800DT$HuVWBo
FOR-1.20  800DT$HuVWBn
20 entries were displayed.
Cluster01::>

```

외부 스토리지 **LUN**을 가져오기 후보로 등록합니다

외부 LUN은 처음에 특정 LUN 유형으로 분류됩니다. 데이터를 가져오려면 먼저 LUN에 외부 태그가 지정되어야 하므로 가져오기 프로세스의 대상이 되어야 합니다. 이 단계는 에 일련 번호를 전달하여 완료합니다 storage disk modify 명령을 사용합니다. 이 프로세스에서는 ONTAP 내의 LUN에만 Foreign으로 태그를 지정합니다. 외부 LUN 자체에 데이터가 기록되지 않습니다.

```

Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBW} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBX} -is
-foreign true
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBn} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBo} -is
-foreign true
Cluster01::*>

```

마이그레이션된 LUN을 호스팅할 볼륨을 생성합니다

마이그레이션된 LUN을 호스팅하려면 볼륨이 필요합니다. 정확한 볼륨 구성은 ONTAP 기능을 활용하는 전체 계획에 따라 다릅니다. 이 예에서는 ASM LUN이 한 볼륨에 배치되고 LVM LUN은 두 번째 볼륨에 배치됩니다. 따라서 계층화, 스냅샷 생성 또는 QoS 제어 설정과 같은 목적으로 LUN을 독립 그룹으로 관리할 수 있습니다.

를 설정합니다 snapshot-policy `to` none. 마이그레이션 프로세스에는 많은 데이터 회전율이 포함될 수 있습니다. 따라서 원하지 않는 데이터가 스냅샷에 캡처되기 때문에 실수로 스냅샷을 생성하는 경우 공간 소비가 크게 증가할 수 있습니다.

```
Cluster01::> volume create -volume new_asm -aggregate data_02 -size 120G
-snapshot-policy none
[Job 1152] Job succeeded: Successful
Cluster01::> volume create -volume new_lvm -aggregate data_02 -size 120G
-snapshot-policy none
[Job 1153] Job succeeded: Successful
Cluster01::>
```

ONTAP LUN을 생성합니다

볼륨을 생성한 후에는 새 LUN을 생성해야 합니다. 일반적으로 LUN을 생성할 때 사용자가 LUN 크기와 같은 정보를 지정해야 하지만 이 경우에는 외부 디스크 인수가 명령에 전달됩니다. 따라서 ONTAP는 지정된 일련 번호에서 현재 LUN 구성 데이터를 복제합니다. 또한 LUN 형태 및 파티션 테이블 데이터를 사용하여 LUN 정렬을 조정하고 최적의 성능을 설정합니다.

이 단계에서는 올바른 외부 LUN이 올바른 새 LUN과 일치하는지 확인하기 위해 외부 스토리지와 일련 번호를 상호 참조해야 합니다.

```
Cluster01::~*> lun create -vserver vserver1 -path /vol/new_asm/LUN0 -ostype
linux -foreign-disk 800DT$HuVWBW
Created a LUN of size 10g (10737418240)
Cluster01::~*> lun create -vserver vserver1 -path /vol/new_asm/LUN1 -ostype
linux -foreign-disk 800DT$HuVWBX
Created a LUN of size 10g (10737418240)
...
Created a LUN of size 10g (10737418240)
Cluster01::~*> lun create -vserver vserver1 -path /vol/new_lvm/LUN8 -ostype
linux -foreign-disk 800DT$HuVWBn
Created a LUN of size 10g (10737418240)
Cluster01::~*> lun create -vserver vserver1 -path /vol/new_lvm/LUN9 -ostype
linux -foreign-disk 800DT$HuVWBo
Created a LUN of size 10g (10737418240)
```

가져오기 관계를 만듭니다

이제 LUN이 생성되었지만 복제 대상으로 구성되지 않았습니다. 이 단계를 수행하려면 먼저 LUN을 오프라인 상태로 전환해야 합니다. 이 추가 단계는 사용자 오류로부터 데이터를 보호하도록 설계되었습니다. ONTAP에서 온라인 LUN에

대해 마이그레이션을 수행할 수 있었다면 인쇄 오류로 인해 활성 데이터를 덮어쓸 위험이 발생할 수 있습니다. 사용자가 먼저 LUN을 오프라인으로 전환하도록 하는 추가 단계는 올바른 타겟 LUN이 마이그레이션 대상으로 사용되는지 확인하는 데 도움이 됩니다.

```
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_asm/LUN0
Warning: This command will take LUN "/vol/new_asm/LUN0" in Vserver
        "vserver1" offline.
Do you want to continue? {y|n}: y
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_asm/LUN1
Warning: This command will take LUN "/vol/new_asm/LUN1" in Vserver
        "vserver1" offline.
Do you want to continue? {y|n}: y
...
Warning: This command will take LUN "/vol/new_lvm/LUN8" in Vserver
        "vserver1" offline.
Do you want to continue? {y|n}: y
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_lvm/LUN9
Warning: This command will take LUN "/vol/new_lvm/LUN9" in Vserver
        "vserver1" offline.
Do you want to continue? {y|n}: y
```

LUN이 오프라인 상태가 된 후 외부 LUN 일련 번호를 에 전달하여 임포트 관계를 설정할 수 있습니다 `lun import create` 명령.

```
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_asm/LUN0
-foreign-disk 800DT$HuVWBW
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_asm/LUN1
-foreign-disk 800DT$HuVWBX
...
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_lvm/LUN8
-foreign-disk 800DT$HuVWBn
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_lvm/LUN9
-foreign-disk 800DT$HuVWBo
Cluster01::*>
```

모든 임포트 관계가 설정되면 LUN을 다시 온라인 상태로 전환할 수 있습니다.

```
Cluster01::*> lun online -vserver vserver1 -path /vol/new_asm/LUN0
Cluster01::*> lun online -vserver vserver1 -path /vol/new_asm/LUN1
...
Cluster01::*> lun online -vserver vserver1 -path /vol/new_lvm/LUN8
Cluster01::*> lun online -vserver vserver1 -path /vol/new_lvm/LUN9
Cluster01::*>
```

이니시에이터 그룹을 생성합니다

igroup(이니시에이터 그룹)은 ONTAP LUN 마스크 아키텍처의 일부입니다. 호스트에 처음으로 액세스 권한이 부여되지 않으면 새로 생성된 LUN에 액세스할 수 없습니다. 이 작업은 액세스 권한을 부여해야 할 FC WWN 또는 iSCSI 이니시에이터 이름을 나열하는 igroup을 생성하여 수행합니다. 이 보고서가 작성된 시점을 기준으로 FLI는 FC LUN에 대해서만 지원됩니다. 그러나 예 나와 있는 것처럼 iSCSI 사후 마이그레이션으로 변환하는 작업은 간단합니다 "[프로토콜 변환](#)".

이 예에서는 호스트의 HBA에서 사용 가능한 두 포트에 해당하는 두 개의 WWN이 포함된 igroup을 생성합니다.

```
Cluster01::*> igroup create linuxhost -protocol fcp -ostype linux
-initiator 21:00:00:0e:1e:16:63:50 21:00:00:0e:1e:16:63:51
```

호스트에 새 LUN 매핑

igroup 작성 후에 LUN이 정의된 igroup에 매핑됩니다. 이 LUN은 이 igroup에 포함된 WWN에만 사용할 수 있습니다. NetApp는 마이그레이션 프로세스에서 이 단계에서 호스트가 ONTAP에 조닝되지 않은 것으로 가정합니다. 이는 호스트가 외부 스토리지와 새 ONTAP 시스템에 동시에 조닝되는 경우 동일한 일련 번호를 가진 LUN이 각 어레이에서 검색될 위험이 있기 때문에 중요합니다. 이 상황은 다중 경로 오작동이나 데이터 손상으로 이어질 수 있습니다.

```
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxhost
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxhost
...
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxhost
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxhost
Cluster01::*>
```

FLI-컷오버를 사용한 Oracle 마이그레이션

FC 네트워크 구성을 변경해야 하기 때문에 외부 LUN 임포트 중에는 중단이 불가피합니다. 그러나 운영 중단은 데이터베이스 환경을 재시작하고 FC 조닝을 업데이트하여 호스트 FC 연결을 외부 LUN에서 ONTAP로 전환하는 데 필요한 시간보다 훨씬 오래 지속되지 않습니다.

이 프로세스는 다음과 같이 요약할 수 있습니다.

1. 외부 LUN에서 모든 LUN 작업을 중지합니다.
2. 호스트 FC 연결을 새 ONTAP 시스템으로 리디렉션합니다.
3. 가져오기 프로세스를 트리거합니다.
4. LUN을 다시 검색합니다.
5. 데이터베이스를 다시 시작합니다.

마이그레이션 프로세스가 완료될 때까지 기다리지 않아도 됩니다. 특정 LUN의 마이그레이션이 시작되는 즉시 ONTAP에서 사용할 수 있으며 데이터 복사 프로세스가 진행되는 동안 데이터를 제공할 수 있습니다. 모든 읽기는 외부 LUN으로 전달되고 모든 쓰기는 두 스토리지에 동기식으로 기록됩니다. 복사 작업은 매우 빠르고 FC 트래픽 리디렉션의 오버헤드가 최소화되므로 성능에 미치는 영향은 일시적이고 최소화해야 합니다. 문제가 있는 경우 마이그레이션 프로세스가 완료되고 가져오기 관계가 삭제될 때까지 환경 다시 시작을 지연시킬 수 있습니다.

데이터베이스를 종료합니다

이 예에서 환경을 정지하는 첫 번째 단계는 데이터베이스를 종료하는 것입니다.

```
[oracle@host1 bin]$ . oraenv
ORACLE_SID = [oracle] ? FLIDB
The Oracle base remains unchanged with value /orabin
[oracle@host1 bin]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, Automatic Storage Management, OLAP, Advanced
Analytics
and Real Application Testing options
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

그리드 서비스를 종료합니다

마이그레이션되는 SAN 기반 파일 시스템 중 하나에 Oracle ASM 서비스도 포함됩니다. 기본 LUN을 정지하려면 파일 시스템을 마운트 해제해야 합니다. 즉, 이 파일 시스템에서 열려 있는 파일이 있는 프로세스를 모두 중지해야 합니다.

```
[oracle@host1 bin]$ ./crsctl stop has -f
CRS-2791: Starting shutdown of Oracle High Availability Services-managed
resources on 'host1'
CRS-2673: Attempting to stop 'ora.evmd' on 'host1'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host1'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host1'
CRS-2677: Stop of 'ora.DATA.dg' on 'host1' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host1'
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host1' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host1' succeeded
CRS-2677: Stop of 'ora.asm' on 'host1' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host1'
CRS-2677: Stop of 'ora.cssd' on 'host1' succeeded
CRS-2793: Shutdown of Oracle High Availability Services-managed resources
on 'host1' has completed
CRS-4133: Oracle High Availability Services has been stopped.
[oracle@host1 bin]$
```

파일 시스템을 마운트 해제합니다

모든 프로세스가 종료되면 마운트 해제 작업이 성공합니다. 사용 권한이 거부되면 파일 시스템에 잠금이 설정된 프로세스가 있어야 합니다. 를 클릭합니다 `fuser` 명령은 이러한 프로세스를 식별하는 데 도움이 될 수 있습니다.

```
[root@host1 ~]# umount /orabin
[root@host1 ~]# umount /backups
```

볼륨 그룹을 비활성화합니다

지정된 볼륨 그룹의 모든 파일 시스템이 마운트 해제된 후 볼륨 그룹을 비활성화할 수 있습니다.

```
[root@host1 ~]# vgchange --activate n sanvg
  0 logical volume(s) in volume group "sanvg" now active
[root@host1 ~]#
```

FC 네트워크 변경 사항

이제 FC 존을 업데이트하여 호스트에서 외부 스토리지에 대한 모든 액세스를 제거하고 ONTAP에 대한 액세스를 설정할 수 있습니다.

가져오기 프로세스를 시작합니다

LUN 가져오기 프로세스를 시작하려면 를 실행합니다 `lun import start` 명령.

```

Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_asm/LUN0
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_asm/LUN1
...
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_lvm/LUN8
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_lvm/LUN9
Cluster01::lun import*>

```

가져오기 진행 상황을 모니터링합니다

를 사용하여 가져오기 작업을 모니터링할 수 있습니다 lun import show 명령. 아래와 같이 20개의 LUN을 모두 가져오는 작업이 진행 중입니다. 따라서 이제 데이터 복사 작업이 계속 진행되더라도 ONTAP를 통해 데이터에 액세스할 수 있습니다.

```

Cluster01::lun import*> lun import show -fields path,percent-complete
vserver    foreign-disk path                percent-complete
-----
vserver1   800DT$HuVWB/ /vol/new_asm/LUN4 5
vserver1   800DT$HuVWBW /vol/new_asm/LUN0 5
vserver1   800DT$HuVWBX /vol/new_asm/LUN1 6
vserver1   800DT$HuVWBZ /vol/new_asm/LUN2 6
vserver1   800DT$HuVWBZ /vol/new_asm/LUN3 5
vserver1   800DT$HuVWBa /vol/new_asm/LUN5 4
vserver1   800DT$HuVWBb /vol/new_asm/LUN6 4
vserver1   800DT$HuVWBc /vol/new_asm/LUN7 4
vserver1   800DT$HuVWBd /vol/new_asm/LUN8 4
vserver1   800DT$HuVWBe /vol/new_asm/LUN9 4
vserver1   800DT$HuVWBf /vol/new_lvm/LUN0 5
vserver1   800DT$HuVWBg /vol/new_lvm/LUN1 4
vserver1   800DT$HuVWBh /vol/new_lvm/LUN2 4
vserver1   800DT$HuVWBj /vol/new_lvm/LUN3 3
vserver1   800DT$HuVWBk /vol/new_lvm/LUN4 3
vserver1   800DT$HuVWBk /vol/new_lvm/LUN5 3
vserver1   800DT$HuVWBk /vol/new_lvm/LUN6 4
vserver1   800DT$HuVWBm /vol/new_lvm/LUN7 3
vserver1   800DT$HuVWBn /vol/new_lvm/LUN8 2
vserver1   800DT$HuVWBo /vol/new_lvm/LUN9 2
20 entries were displayed.

```

오프라인 프로세스가 필요한 경우 까지 서비스 재검색 또는 재시작을 연기합니다 lun import show 명령은 모든 마이그레이션이 성공적이고 완료되었음을 나타냅니다. 그런 다음 에 설명된 대로 마이그레이션 프로세스를 완료할 수 있습니다 ["외부 LUN 임포트 - 완료"](#).

온라인 마이그레이션이 필요한 경우 새 집에서 LUN을 다시 검색하고 서비스를 시작합니다.

SCSI 장치 변경 사항을 검색합니다

대부분의 경우 새 LUN을 다시 검색하는 가장 간단한 옵션은 호스트를 재시작하는 것입니다. 이렇게 하면 오래된 오래된 장치가 자동으로 제거되고 모든 새 LUN이 올바르게 검색되며 다중 경로 장치와 같은 관련 장치가 구축됩니다. 이 예제에서는 데모를 위한 완전한 온라인 프로세스를 보여 줍니다.

주의: 호스트를 다시 시작하기 전에 의 모든 항목이 있는지 확인하십시오 `/etc/fstab` 마이그레이션된 참조 SAN 리소스가 주석 처리되었습니다. 이렇게 하지 않고 LUN 액세스에 문제가 있으면 운영 체제가 부팅되지 않을 수 있습니다. 이 상황은 데이터를 손상시키지 않습니다. 그러나 구조 모드 또는 유사한 모드로 부팅하고 를 수정하는 것은 매우 불편할 수 있습니다 `/etc/fstab` 문제 해결을 위해 운영 체제를 부팅할 수 있습니다.

이 예에 사용된 Linux 버전의 LUN을 로 다시 검색할 수 있습니다 `rescan-scsi-bus.sh` 명령. 명령이 성공하면 각 LUN 경로가 출력에 표시되어야 합니다. 출력에서 해석하기가 어려울 수 있지만, 조닝 및 `igroup` 구성이 올바르게 을 포함하는 많은 LUN이 표시되어야 합니다 NETAPP 공급업체 문자열

```

[root@host1 /]# rescan-scsi-bus.sh
Scanning SCSI subsystem for new devices
Scanning host 0 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
  Scanning for device 0 2 0 0 ...
OLD: Host: scsi0 Channel: 02 Id: 00 Lun: 00
      Vendor: LSI      Model: RAID SAS 6G 0/1  Rev: 2.13
      Type:   Direct-Access                    ANSI SCSI revision: 05
Scanning host 1 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
  Scanning for device 1 0 0 0 ...
OLD: Host: scsi1 Channel: 00 Id: 00 Lun: 00
      Vendor: Optiarc  Model: DVD RW AD-7760H  Rev: 1.41
      Type:   CD-ROM                      ANSI SCSI revision: 05
Scanning host 2 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
Scanning host 3 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
Scanning host 4 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
Scanning host 5 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
Scanning host 6 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
Scanning host 7 for all SCSI target IDs, all LUNs
  Scanning for device 7 0 0 10 ...
OLD: Host: scsi7 Channel: 00 Id: 00 Lun: 10
      Vendor: NETAPP   Model: LUN C-Mode      Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
  Scanning for device 7 0 0 11 ...
OLD: Host: scsi7 Channel: 00 Id: 00 Lun: 11
      Vendor: NETAPP   Model: LUN C-Mode      Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
  Scanning for device 7 0 0 12 ...
...
OLD: Host: scsi9 Channel: 00 Id: 01 Lun: 18
      Vendor: NETAPP   Model: LUN C-Mode      Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
  Scanning for device 9 0 1 19 ...
OLD: Host: scsi9 Channel: 00 Id: 01 Lun: 19
      Vendor: NETAPP   Model: LUN C-Mode      Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
0 new or changed device(s) found.
0 remapped or resized device(s) found.
0 device(s) removed.

```

다중 경로 장치를 확인합니다

LUN 검색 프로세스는 다중 경로 장치의 재구성을 트리거하지만 Linux 다중 경로 드라이버에는 간헐적인 문제가 있는 것으로 알려져 있습니다. 의 출력입니다 `multipath - ll` 출력이 예상한 대로 나타나는지 확인해야 합니다. 예를 들어, 아래 출력에는 와 연결된 다중 경로 장치가 나와 있습니다 NETAPP 공급업체 문자열 각 디바이스에는 4개의 경로가 있으며, 우선 순위가 50이고 우선 순위가 10인 2개의 경로가 있습니다. 정확한 출력은 Linux 버전에 따라 다를 수 있지만 이 출력은 예상한 대로 표시됩니다.



사용하는 Linux 버전에 대한 호스트 유틸리티 설명서를 참조하여 를 확인하십시오
/etc/multipath.conf 설정이 올바릅니다.

```
[root@host1 /]# multipath -ll
3600a098038303558735d493762504b36 dm-5 NETAPP ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:4 sdat 66:208 active ready running
| `-- 9:0:1:4 sdbn 68:16 active ready running
`+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:4 sdf 8:80 active ready running
  `-- 9:0:0:4 sdz 65:144 active ready running
3600a098038303558735d493762504b2d dm-10 NETAPP ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:8 sdax 67:16 active ready running
| `-- 9:0:1:8 sdbn 68:80 active ready running
`+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:8 sdj 8:144 active ready running
  `-- 9:0:0:8 sdad 65:208 active ready running
...
3600a098038303558735d493762504b37 dm-8 NETAPP ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:5 sdau 66:224 active ready running
| `-- 9:0:1:5 sdbo 68:32 active ready running
`+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:5 sdg 8:96 active ready running
  `-- 9:0:0:5 sdaa 65:160 active ready running
3600a098038303558735d493762504b4b dm-22 NETAPP ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:19 sdbi 67:192 active ready running
| `-- 9:0:1:19 sdcc 69:0 active ready running
`+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:19 sdu 65:64 active ready running
  `-- 9:0:0:19 sdao 66:128 active ready running
```

LVM 볼륨 그룹을 다시 활성화합니다

LVM LUN이 제대로 검색되면 가 나타납니다 vgchange --activate y 명령이 성공해야 합니다. 이것은 논리적

볼륨 관리자의 값에 대한 좋은 예입니다. 볼륨 그룹 메타데이터가 LUN 자체에 기록되므로 LUN의 WWN이나 일련 번호도 변경하는 것은 중요하지 않습니다.

OS는 LUN을 검색하여 LUN에 기록된 소량의 데이터를 발견했습니다. 이 데이터는 LUN에 속한 물리적 볼륨으로 식별됩니다 `sanvg volumegroup`. 그런 다음 필요한 모든 장치를 구축했습니다. 볼륨 그룹을 다시 활성화하기만 하면 됩니다.

```
[root@host1 ~]# vgchange --activate y sanvg
Found duplicate PV fpCzdLTuKfy2xDZjailNliJh3TjLUBiT: using
/dev/mapper/3600a098038303558735d493762504b46 not /dev/sdp
Using duplicate PV /dev/mapper/3600a098038303558735d493762504b46 from
subsystem DM, ignoring /dev/sdp
2 logical volume(s) in volume group "sanvg" now active
```

파일 시스템을 다시 마운트합니다

볼륨 그룹이 다시 활성화된 후 모든 원본 데이터가 손상되지 않은 상태로 파일 시스템을 마운트할 수 있습니다. 앞서 설명했듯이 백 그룹에서 데이터 복제가 아직 활성 상태인 경우에도 파일 시스템이 완전히 작동합니다.

```
[root@host1 ~]# mount /orabin
[root@host1 ~]# mount /backups
[root@host1 ~]# df -k
```

Filesystem	1K-blocks	Used	Available	Use%
Mounted on				
/dev/mapper/rhel-root	52403200	8837100	43566100	17% /
devtmpfs	65882776	0	65882776	0% /dev
tmpfs	6291456	84	6291372	1%
/dev/shm				
tmpfs	65898668	9884	65888784	1% /run
tmpfs	65898668	0	65898668	0%
/sys/fs/cgroup				
/dev/sda1	505580	224828	280752	45% /boot
fas8060-nfs-public:/install	199229440	119368256	79861184	60%
/install				
fas8040-nfs-routable:/snapomatic	9961472	30528	9930944	1%
/snapomatic				
tmpfs	13179736	16	13179720	1%
/run/user/42				
tmpfs	13179736	0	13179736	0%
/run/user/0				
/dev/mapper/sanvg-lvorabin	20961280	12357456	8603824	59%
/orabin				
/dev/mapper/sanvg-lvbackups	73364480	62947536	10416944	86%
/backups				

ASM 장치를 다시 검색합니다

SCSI 장치를 다시 검색할 때 ASMLib 장치를 다시 검색해야 합니다. ASMLib를 다시 시작한 다음 디스크를 검사하여 온라인으로 재검색을 확인할 수 있습니다.



이 단계는 ASMLib가 사용되는 ASM 구성에만 관련이 있습니다.

주의: ASMLib를 사용하지 않는 경우 /dev/mapper 디바이스가 자동으로 다시 생성되어야 합니다. 그러나 사용 권한이 올바르지 않을 수 있습니다. ASMLib가 없는 경우 ASM에 대한 기본 장치에 특수 권한을 설정해야 합니다. 이러한 작업은 일반적으로 둘 중 하나의 특수 항목을 통해 수행됩니다 /etc/multipath.conf 또는 udev 두 규칙 집합에서 사용할 수 있습니다. 이러한 파일은 WWN 또는 일련 번호 측면에서 환경의 변경 사항을 반영하도록 업데이트하여 ASM 장치에 올바른 권한이 있는지 확인해야 할 수 있습니다.

이 예에서는 ASMLib를 다시 시작하고 디스크 검색을 통해 원래 환경과 동일한 10개의 ASM LUN을 표시합니다.

```
[root@host1 ~]# oracleasm exit
Unmounting ASMLib driver filesystem: /dev/oracleasm
Unloading module "oracleasm": oracleasm
[root@host1 ~]# oracleasm init
Loading module "oracleasm": oracleasm
Configuring "oracleasm" to use device physical block size
Mounting ASMLib driver filesystem: /dev/oracleasm
[root@host1 ~]# oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "ASM0"
Instantiating disk "ASM1"
Instantiating disk "ASM2"
Instantiating disk "ASM3"
Instantiating disk "ASM4"
Instantiating disk "ASM5"
Instantiating disk "ASM6"
Instantiating disk "ASM7"
Instantiating disk "ASM8"
Instantiating disk "ASM9"
```

그리드 서비스를 다시 시작합니다

이제 LVM 및 ASM 장치가 온라인 상태이고 사용 가능해졌으므로 그리드 서비스를 다시 시작할 수 있습니다.

```
[root@host1 ~]# cd /orabin/product/12.1.0/grid/bin
[root@host1 bin]# ./crsctl start has
```

데이터베이스를 다시 시작합니다

그리드 서비스가 다시 시작된 후 데이터베이스를 불러올 수 있습니다. 데이터베이스를 시작하기 전에 ASM 서비스를 완전히 사용할 수 있도록 몇 분 정도 기다려야 할 수 있습니다.

```
[root@host1 bin]# su - oracle
[oracle@host1 ~]$ . oraenv
ORACLE_SID = [oracle] ? FLIDB
The Oracle base has been set to /orabin
[oracle@host1 ~]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to an idle instance.
SQL> startup
ORACLE instance started.
Total System Global Area 3221225472 bytes
Fixed Size 4502416 bytes
Variable Size 1207962736 bytes
Database Buffers 1996488704 bytes
Redo Buffers 12271616 bytes
Database mounted.
Database opened.
SQL>
```

FLI를 사용한 **Oracle** 마이그레이션 완료

호스트 관점에서 보면 마이그레이션이 완료되지만 가져오기 관계가 삭제될 때까지 외부 스토리지에서 입출력이 계속 제공됩니다.

관계를 삭제하기 전에 모든 LUN에 대해 마이그레이션 프로세스가 완료되었는지 확인해야 합니다.

```

Cluster01::*> lun import show -vserver vserver1 -fields foreign-
disk,path,operational-state
vserver    foreign-disk path                operational-state
-----
vserver1 800DT$HuVWB/ /vol/new_asm/LUN4 completed
vserver1 800DT$HuVWBW /vol/new_asm/LUN0 completed
vserver1 800DT$HuVWBX /vol/new_asm/LUN1 completed
vserver1 800DT$HuVWBZ /vol/new_asm/LUN2 completed
vserver1 800DT$HuVWBa /vol/new_asm/LUN5 completed
vserver1 800DT$HuVWBb /vol/new_asm/LUN6 completed
vserver1 800DT$HuVWBc /vol/new_asm/LUN7 completed
vserver1 800DT$HuVWBd /vol/new_asm/LUN8 completed
vserver1 800DT$HuVWBe /vol/new_asm/LUN9 completed
vserver1 800DT$HuVWBf /vol/new_lvm/LUN0 completed
vserver1 800DT$HuVWBg /vol/new_lvm/LUN1 completed
vserver1 800DT$HuVWBh /vol/new_lvm/LUN2 completed
vserver1 800DT$HuVWBi /vol/new_lvm/LUN3 completed
vserver1 800DT$HuVWBj /vol/new_lvm/LUN4 completed
vserver1 800DT$HuVWBk /vol/new_lvm/LUN5 completed
vserver1 800DT$HuVWBl /vol/new_lvm/LUN6 completed
vserver1 800DT$HuVWBm /vol/new_lvm/LUN7 completed
vserver1 800DT$HuVWBn /vol/new_lvm/LUN8 completed
vserver1 800DT$HuVWBo /vol/new_lvm/LUN9 completed
20 entries were displayed.

```

관계 가져오기를 삭제합니다

마이그레이션 프로세스가 완료되면 마이그레이션 관계를 삭제합니다. I/O를 완료한 후에는 ONTAP의 드라이브에서만 I/O를 처리합니다.

```

Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_asm/LUN0
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_asm/LUN1
...
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_lvm/LUN8
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_lvm/LUN9

```

외부 LUN 등록을 취소합니다

마지막으로 디스크를 수정하여 를 제거합니다 is-foreign 지정.

```

Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBW} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVVBX} -is
-foreign false
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVVBn} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBo} -is
-foreign false
Cluster01::*>

```

FLI-프로토콜 변환을 통한 Oracle 마이그레이션

LUN에 액세스하는 데 사용되는 프로토콜을 변경하는 것은 일반적인 요구사항입니다.

데이터를 클라우드로 마이그레이션하는 전체 전략의 일부이기도 한 경우도 있습니다. TCP/IP는 클라우드의 프로토콜이며 FC에서 iSCSI로 변경하면 다양한 클라우드 환경으로 쉽게 마이그레이션할 수 있습니다. 그렇지 않으면 iSCSI가 IP SAN의 감소된 비용을 활용하는 것이 바람직할 수도 있습니다. 경우에 따라 마이그레이션이 임시 조치로 다른 프로토콜을 사용할 수 있습니다. 예를 들어, 외부 스토리지 시스템과 ONTAP 기반 LUN이 동일한 HBA에 공존할 수 없는 경우 기존 스토리지의 데이터를 복제할 수 있을 정도로 긴 iSCSI LUN을 사용할 수 있습니다. 그런 다음 이전 LUN을 시스템에서 제거한 후 FC로 다시 변환할 수 있습니다.

다음 절차는 FC에서 iSCSI로 변환하는 방법을 보여 주지만 전반적인 원칙은 역방향 iSCSI에서 FC로 변환하는 방법에 적용됩니다.

iSCSI 이니시에이터를 설치합니다

대부분의 운영 체제에는 기본적으로 소프트웨어 iSCSI 초기자가 포함되어 있지만 포함되어 있지 않은 경우 쉽게 설치할 수 있습니다.

```

[root@host1 /]# yum install -y iscsi-initiator-utils
Loaded plugins: langpacks, product-id, search-disabled-repos,
subscription-
                : manager
Resolving Dependencies
--> Running transaction check
---> Package iscsi-initiator-utils.x86_64 0:6.2.0.873-32.e17 will be
updated
--> Processing Dependency: iscsi-initiator-utils = 6.2.0.873-32.e17 for
package: iscsi-initiator-utils-iscsiuio-6.2.0.873-32.e17.x86_64
---> Package iscsi-initiator-utils.x86_64 0:6.2.0.873-32.0.2.e17 will be
an update
--> Running transaction check
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.e17 will
be updated
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.e17

```

will be an update

--> Finished Dependency Resolution

Dependencies Resolved

=====
===

Package	Arch	Version	Repository
---------	------	---------	------------

=====
===

Updating:

iscsi-initiator-utils	x86_64	6.2.0.873-32.0.2.el7	ol7_latest	416 k
-----------------------	--------	----------------------	------------	----------

Updating for dependencies:

iscsi-initiator-utils-iscsiuio	x86_64	6.2.0.873-32.0.2.el7	ol7_latest	84 k
--------------------------------	--------	----------------------	------------	---------

Transaction Summary

=====
===

Upgrade 1 Package (+1 Dependent package)

Total download size: 501 k

Downloading packages:

No Presto metadata available for ol7_latest

(1/2): iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_6 | 416 kB 00:00

(2/2): iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2. | 84 kB 00:00

Total 2.8 MB/s | 501 kB

00:00Cluster01

Running transaction check

Running transaction test

Transaction test succeeded

Running transaction

Updating : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2.el7.x86
1/4

Updating : iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_64
2/4

Cleanup : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
3/4

Cleanup : iscsi-initiator-utils-6.2.0.873-32.el7.x86_64
4/4

rhel-7-server-eus-rpms/7Server/x86_64/productid | 1.7 kB 00:00

rhel-7-server-rpms/7Server/x86_64/productid | 1.7 kB 00:00

Verifying : iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_64
1/4

Verifying : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2.el7.x86
2/4

```

Verifying   : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
3/4
Verifying   : iscsi-initiator-utils-6.2.0.873-32.el7.x86_64
4/4
Updated:
  iscsi-initiator-utils.x86_64 0:6.2.0.873-32.0.2.el7
Dependency Updated:
  iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.el7
Complete!
[root@host1 ~]#

```

iSCSI 이니시에이터 이름을 식별합니다

설치 프로세스 중에 고유한 iSCSI 이니시에이터 이름이 생성됩니다. Linux에서는 에 있습니다
 /etc/iscsi/initiatorname.iscsi 파일. 이 이름은 IP SAN에서 호스트를 식별하는 데 사용됩니다.

```

[root@host1 ~]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1992-05.com.redhat:497bd66ca0

```

새 이니시에이터 그룹을 생성합니다

igroup(이니시에이터 그룹)은 ONTAP LUN 마스킹 아키텍처의 일부입니다. 호스트에 처음으로 액세스 권한이 부여되지 않으면 새로 생성된 LUN에 액세스할 수 없습니다. 이 단계는 액세스가 필요한 FC WWN 또는 iSCSI 이니시에이터 이름을 나열하는 igroup을 생성하여 수행합니다.

이 예에서는 Linux 호스트의 iSCSI 이니시에이터가 포함된 igroup이 생성됩니다.

```

Cluster01::*> igroup create -igroup linuxiscsi -protocol iscsi -ostype
linux -initiator iqn.1994-05.com.redhat:497bd66ca0

```

환경을 종료합니다

LUN 프로토콜을 변경하기 전에 LUN을 완전히 정지해야 합니다. 변환 중인 LUN 중 하나의 데이터베이스를 종료하고 파일 시스템을 마운트 해제해야 하며 볼륨 그룹을 비활성화해야 합니다. ASM이 사용되는 경우 ASM 디스크 그룹이 분리되고 모든 그리드 서비스가 종료되는지 확인합니다.

FC 네트워크에서 **LUN** 매핑을 해제합니다

LUN이 완전히 정지된 후 원본 FC igroup에서 매핑을 제거합니다.

```
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxhost
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxhost
...
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxhost
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxhost
```

LUN을 IP 네트워크에 다시 매핑합니다

새 iSCSI 기반 이니시에이터 그룹에 각 LUN에 대한 액세스 권한을 부여합니다.

```
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxiscsi
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxiscsi
...
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxiscsi
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxiscsi
Cluster01::*>
```

iSCSI 대상을 검색합니다

iSCSI 검색에는 두 단계가 있습니다. 첫 번째는 LUN 검색과 다른 타겟을 검색하는 것입니다. 를 클릭합니다 `iscsiadm` 아래 표시된 명령은 에서 지정한 포털 그룹을 검색합니다 `-p` argument 및 는 iSCSI 서비스를 제공하는 모든 IP 주소 및 포트의 목록을 저장합니다. 이 경우 기본 포트 3260에 iSCSI 서비스가 있는 네 개의 IP 주소가 있습니다.



대상 IP 주소에 연결할 수 없는 경우 이 명령을 완료하는 데 몇 분 정도 걸릴 수 있습니다.

```
[root@host1 ~]# iscsiadm -m discovery -t st -p fas8060-iscsi-public1
10.63.147.197:3260,1033 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
10.63.147.198:3260,1034 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
172.20.108.203:3260,1030 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
172.20.108.202:3260,1029 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
```

iSCSI LUN을 검색합니다

iSCSI 대상이 검색된 후 iSCSI 서비스를 다시 시작하여 사용 가능한 iSCSI LUN을 검색하고 다중 경로 또는 ASMLib 디바이스와 같은 관련 디바이스를 구축합니다.

```
[root@host1 ~]# service iscsi restart
Redirecting to /bin/systemctl restart iscsi.service
```

환경을 다시 시작합니다

볼륨 그룹을 다시 활성화하고, 파일 시스템을 다시 마운트하고, RAC 서비스를 다시 시작하는 등의 방법으로 환경을 다시 시작합니다. 예방 조치로, NetApp 변환 프로세스가 완료된 후 서버를 재부팅하여 모든 구성 파일이 올바르게 오래된 모든 디바이스가 제거되도록 하는 것이 좋습니다.

주의: 호스트를 다시 시작하기 전에 의 모든 항목이 있는지 확인하십시오 /etc/fstab 마이그레이션된 참조 SAN 리소스가 주석 처리되었습니다. 이 단계를 수행하지 않고 LUN 액세스에 문제가 있는 경우 운영 체제가 부팅되지 않을 수 있습니다. 이 문제는 데이터를 손상시키지 않습니다. 그러나 구조 모드 또는 유사한 모드로 부팅하고 수정하는 것은 매우 불편할 수 있습니다 /etc/fstab 문제 해결 노력을 시작할 수 있도록 운영 체제를 부팅할 수 있습니다.

Oracle 마이그레이션 절차 샘플 스크립트

제공되는 스크립트는 다양한 OS 및 데이터베이스 작업을 스크립팅하는 방법의 예로 제공됩니다. 그들은 있는 그대로 제공됩니다. 특정 절차에 대한 지원이 필요한 경우 NetApp 또는 NetApp 리셀러에게 문의하십시오.

데이터베이스 종료

다음 Perl 스크립트는 Oracle SID의 단일 인수를 사용하고 데이터베이스를 종료합니다. Oracle 사용자 또는 루트로 실행할 수 있습니다.

```

#!/usr/bin/perl
use strict;
use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
my $uid=$<;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
77 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
shutdown immediate;
EOF2
`
`;}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF4
sqlplus / as sysdba << EOF2
shutdown immediate;
EOF2
`;};
print @out;
if ("@out" =~ /ORACLE instance shut down/) {
print "$oraclesid shut down\n";
exit 0;}
elsif ("@out" =~ /Connected to an idle instance/) {
print "$oraclesid already shut down\n";
exit 0;}
else {
print "$oraclesid failed to shut down\n";
exit 1;}

```

데이터베이스 시작

다음 Perl 스크립트는 Oracle SID의 단일 인수를 사용하고 데이터베이스를 종료합니다. Oracle 사용자 또는 루트로 실행할 수 있습니다.

```

#!/usr/bin/perl
use strict;
use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
my $uid=$<;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
startup;
EOF2
`
`;}
else {
@out=`. oraenv << EOF3
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
startup;
EOF2
`;};
print @out;
if ("@out" =~ /Database opened/) {
print "$oraclesid started\n";
exit 0;}
elsif ("@out" =~ /cannot start already-running ORACLE/) {
print "$oraclesid already started\n";
exit 1;}
else {
78 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
print "$oraclesid failed to start\n";
exit 1;}

```

파일 시스템을 읽기 전용으로 변환합니다

다음 스크립트는 파일 시스템 인수를 사용하여 읽기 전용으로 마운트 해제 및 다시 마운트하려고 시도합니다. 이렇게 하면 데이터를 복제하기 위해 파일 시스템을 사용할 수 있어야 하지만 우발적인 손상으로부터 보호해야 하는 마이그레이션 프로세스 중에 유용합니다.

```

#!/usr/bin/perl
use strict;
#use warnings;
my $filesystem=$ARGV[0];
my @out=`umount '$filesystem'`;
if ($? == 0) {
    print "$filesystem unmounted\n";
    @out = `mount -o ro '$filesystem'`;
    if ($? == 0) {
        print "$filesystem mounted read-only\n";
        exit 0;}}
else {
    print "Unable to unmount $filesystem\n";
    exit 1;}
print @out;

```

파일 시스템을 교체합니다

다음 스크립트 예제는 파일 시스템 하나를 다른 파일 시스템으로 바꾸는 데 사용됩니다. '/etc/fstab' 파일을 편집하므로 루트로 실행해야 합니다. 이전 파일 시스템과 새 파일 시스템의 심표로 구분된 단일 인수를 사용할 수 있습니다.

1. 파일 시스템을 교체하려면 다음 스크립트를 실행합니다.

```

#!/usr/bin/perl
use strict;
#use warnings;
my $oldfs;
my $newfs;
my @oldfstab;
my @newfstab;
my $source;
my $mountpoint;
my $leftover;
my $oldfstabentry='';
my $newfstabentry='';
my $migratedfstabentry='';
($oldfs, $newfs) = split(',', $ARGV[0]);
open(my $filehandle, '<', '/etc/fstab') or die "Could not open
/etc/fstab\n";
while (my $line = <$filehandle>) {
    chomp $line;
    ($source, $mountpoint, $leftover) = split(/[ , ]/, $line, 3);
    if ($mountpoint eq $oldfs) {
        $oldfstabentry = "#Removed by swap script $source $oldfs $leftover";}
    elsif ($mountpoint eq $newfs) {

```

```

$newfstabentry = "#Removed by swap script $source $newfs $leftover";
$migratedfstabentry = "$source $oldfs $leftover";
else {
  push (@newfstab, "$line\n");}
79 Migration of Oracle Databases to NetApp Storage Systems © 2021
NetApp, Inc. All rights reserved
push (@newfstab, "$oldfstabentry\n");
push (@newfstab, "$newfstabentry\n");
push (@newfstab, "$migratedfstabentry\n");
close($filehandle);
if ($oldfstabentry eq ''){
  die "Could not find $oldfs in /etc/fstab\n";}
if ($newfstabentry eq ''){
  die "Could not find $newfs in /etc/fstab\n";}
my @out=`umount '$newfs'`;
if ($? == 0) {
  print "$newfs unmounted\n";}
else {
  print "Unable to unmount $newfs\n";
  exit 1;}
@out=`umount '$oldfs'`;
if ($? == 0) {
  print "$oldfs unmounted\n";}
else {
  print "Unable to unmount $oldfs\n";
  exit 1;}
system("cp /etc/fstab /etc/fstab.bak");
open ($filehandle, ">", '/etc/fstab') or die "Could not open /etc/fstab
for writing\n";
for my $line (@newfstab) {
  print $filehandle $line;}
close($filehandle);
@out=`mount '$oldfs'`;
if ($? == 0) {
  print "Mounted updated $oldfs\n";
  exit 0;}
else{
  print "Unable to mount updated $oldfs\n";
  exit 1;}
exit 0;

```

이 스크립트 사용의 예로, 의 데이터를 가정합니다 /oradata 로 마이그레이션됩니다 /neworadata 및 /logs 로 마이그레이션됩니다 /newlogs. 이 작업을 수행하는 가장 간단한 방법 중 하나는 간단한 파일 복제 작업을 사용하여 새 디바이스를 원래 마운트 지점으로 재배치하는 것입니다.

2. 예 이전 파일 시스템과 새 파일 시스템이 있다고 가정합니다 /etc/fstab 다음과 같은 파일:

```

cluster01:/vol_oradata /oradata nfs rw,bg,vers=3,rsize=65536,wsiz=65536
0 0
cluster01:/vol_logs /logs nfs rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
cluster01:/vol_newlogs /newlogs nfs rw,bg,vers=3,rsize=65536,wsiz=65536
0 0

```

3. 이 스크립트를 실행하면 현재 파일 시스템을 마운트 해제하고 새 파일 시스템으로 대체합니다.

```

[root@jpsc3 scripts]# ./swap.fs.pl /oradata,/neworadata
/neworadata unmounted
/oradata unmounted
Mounted updated /oradata
[root@jpsc3 scripts]# ./swap.fs.pl /logs,/newlogs
/newlogs unmounted
/logs unmounted
Mounted updated /logs

```

4. 스크립트도 를 업데이트합니다 /etc/fstab 그에 따라 보관합니다. 여기에 표시된 예에서는 다음과 같은 변경 사항이 포함되어 있습니다.

```

#Removed by swap script cluster01:/vol_oradata /oradata nfs
rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
#Removed by swap script cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
cluster01:/vol_neworadata /oradata nfs
rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
#Removed by swap script cluster01:/vol_logs /logs nfs
rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
#Removed by swap script cluster01:/vol_newlogs /newlogs nfs
rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
cluster01:/vol_newlogs /logs nfs rw,bg,vers=3,rsize=65536,wsiz=65536 0
0

```

자동화된 데이터베이스 마이그레이션

이 예제에서는 마이그레이션을 완전히 자동화하기 위해 종료, 시작 및 파일 시스템 교체 스크립트를 사용하는 방법을 보여 줍니다.

```

#!/usr/bin/perl
use strict;

```

```

#use warnings;
my $oraclesid=$ARGV[0];
my @oldfs;
my @newfs;
my $x=1;
while ($x < scalar(@ARGV)) {
    ($oldfs[$x-1], $newfs[$x-1]) = split (',', $ARGV[$x]);
    $x+=1;}
my @out=`./dbshut.pl '$oraclesid'`;
print @out;
if ($? ne 0) {
    print "Failed to shut down database\n";
    exit 0;}
$x=0;
while ($x < scalar(@oldfs)) {
    my @out=`./mk.fs.readonly.pl '$oldfs[$x]'`;
    if ($? ne 0) {
        print "Failed to make filesystem $oldfs[$x] readonly\n";
        exit 0;}
    $x+=1;}
$x=0;
while ($x < scalar(@oldfs)) {
    my @out=`rsync -rlpogt --stats --progress --exclude='.snapshot'
'$oldfs[$x]/' '$newfs[$x]/'`;
    print @out;
    if ($? ne 0) {
        print "Failed to copy filesystem $oldfs[$x] to $newfs[$x]\n";
        exit 0;}
    else {
        print "Succesfully replicated filesystem $oldfs[$x] to
$newfs[$x]\n";}
    $x+=1;}
$x=0;
while ($x < scalar(@oldfs)) {
    print "swap $x $oldfs[$x] $newfs[$x]\n";
    my @out=`./swap.fs.pl '$oldfs[$x], $newfs[$x]'`;
    print @out;
    if ($? ne 0) {
        print "Failed to swap filesystem $oldfs[$x] for $newfs[$x]\n";
        exit 1;}
    else {
        print "Swapped filesystem $oldfs[$x] for $newfs[$x]\n";}
    $x+=1;}
my @out=`./dbstart.pl '$oraclesid'`;
print @out;

```

파일 위치를 표시합니다

이 스크립트는 많은 중요한 데이터베이스 매개 변수를 수집하여 읽기 쉬운 형식으로 인쇄합니다. 이 스크립트는 데이터 레이아웃을 검토할 때 유용할 수 있습니다. 또한 스크립트를 다른 용도로 수정할 수도 있습니다.

```
#!/usr/bin/perl
#use strict;
#use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
sub dosql{
    my $command = @_[0];
    my @lines;
    my $uid=$<;
    if ($uid == 0) {
        @lines=`su - $oracleuser -c "export ORAENV_ASK=NO;export
ORACLE_SID=$oraclesid;. oraenv -s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
"
        `; }
    else {
        $command=~s/\\\\\\\\\\\\\\\\/\\/g;
        @lines=`export ORAENV_ASK=NO;export ORACLE_SID=$oraclesid;. oraenv
-s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
        `; };
    return @lines}
print "\n";
@out=dosql('select name from v\\\\\\\\\\\\\\\\$datafile;');
print "$oraclesid datafiles:\n";
for $line (@out) {
    chomp($line);
    if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('select member from v\\\\\\\\\\\\\\\\$logfile;');
print "$oraclesid redo logs:\n";
for $line (@out) {
```

```

        chomp($line);
        if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('select name from v\\\\\\\\$tempfile;');
print "$oraclesid temp datafiles:\n";
for $line (@out) {
    chomp($line);
    if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('show parameter spfile;');
print "$oraclesid spfile\n";
for $line (@out) {
    chomp($line);
    if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('select name||\'' \'|value from v\\\\\\\\$parameter where
isdefault=\'FALSE\';');
print "$oraclesid key parameters\n";
for $line (@out) {
    chomp($line);
    if ($line =~ /control_files/) {print "$line\n";}
    if ($line =~ /db_create/) {print "$line\n";}
    if ($line =~ /db_file_name_convert/) {print "$line\n";}
    if ($line =~ /log_archive_dest/) {print "$line\n";}}
    if ($line =~ /log_file_name_convert/) {print "$line\n";}
    if ($line =~ /pdb_file_name_convert/) {print "$line\n";}
    if ($line =~ /spfile/) {print "$line\n";}
print "\n";

```

ASM 마이그레이션 정리

```

#!/usr/bin/perl
#use strict;
#use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
sub dosql{
    my $command = @_ [0];
    my @lines;
    my $uid=$<;
    if ($uid == 0) {
        @lines=`su - $oracleuser -c "export ORAENV_ASK=NO;export
ORACLE_SID=$oraclesid;. oraenv -s << EOF1
EOF1

```

```

sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
"
    `; }
    else {
        $command=~s/\\\\\\\\/\\/g;
        @lines=`export ORAENV_ASK=NO;export ORACLE_SID=$oraclesid;. oraenv
-s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
    `; }
return @lines}
print "\n";
@out=dosql('select name from v\\\\\\\\$datafile;');
print @out;
print "shutdown immediate;\n";
print "startup mount;\n";
print "\n";
for $line (@out) {
    if (length($line) > 1) {
        chomp($line);
        ($first, $second,$third,$fourth)=split('_', $line);
        $fourth =~ s/^TS-//;
        $newname=lc("$fourth.dbf");
        $path2file=$line;
        $path2file=~ /(^.*\\.\/)/;
        print "host mv $line $1$newname\n";}}
print "\n";
for $line (@out) {
    if (length($line) > 1) {
        chomp($line);
        ($first, $second,$third,$fourth)=split('_', $line);
        $fourth =~ s/^TS-//;
        $newname=lc("$fourth.dbf");
        $path2file=$line;
        $path2file=~ /(^.*\\.\/)/;
        print "alter database rename file '$line' to
'$1$newname';\n";}}
print "alter database open;\n";
print "\n";

```

ASM에서 파일 시스템 이름으로 변환

```
set serveroutput on;
set wrap off;
declare
    cursor df is select file#, name from v$datafile;
    cursor tf is select file#, name from v$tempfile;
    cursor lf is select member from v$logfile;
    firstline boolean := true;
begin
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('Parameters for log file conversion:');
    dbms_output.put_line(CHR(13));
    dbms_output.put('*.log_file_name_convert = ');
    for lfrec in lf loop
        if (firstline = true) then
            dbms_output.put('''' || lfrec.member || ''', ');
            dbms_output.put(''''/NEW_PATH/' ||
regexp_replace(lfrec.member, '^.*./', '') || ''');
        else
            dbms_output.put(', ''' || lfrec.member || ''', ');
            dbms_output.put(''''/NEW_PATH/' ||
regexp_replace(lfrec.member, '^.*./', '') || ''');
        end if;
        firstline:=false;
    end loop;
    dbms_output.put_line(CHR(13));
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('rman duplication script:');
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('run');
    dbms_output.put_line('{');
    for dfrec in df loop
        dbms_output.put_line('set newname for datafile ' ||
            dfrec.file# || ' to ''' || dfrec.name || ''';');
    end loop;
    for tfrec in tf loop
        dbms_output.put_line('set newname for tempfile ' ||
            tfrec.file# || ' to ''' || tfrec.name || ''';');
    end loop;
    dbms_output.put_line('duplicate target database for standby backup
location INSERT_PATH_HERE;');
    dbms_output.put_line('}');
end;
/
```

데이터베이스에서 로그를 재생합니다

이 스크립트는 마운트 모드에 있는 데이터베이스에 대해 Oracle SID의 단일 인수를 허용하고 현재 사용 가능한 모든 아카이브 로그를 재생하려고 시도합니다.

```
#!/usr/bin/perl
use strict;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
84 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
my $uid = $<;
my @out;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover database until cancel;
auto
EOF2
`;
}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover database until cancel;
auto
EOF2
`;
}
print @out;
```

대기 데이터베이스에서 로그를 재생합니다

이 스크립트는 대기 데이터베이스용으로 설계되었다는 점을 제외하고 위의 스크립트와 동일합니다.

```

#!/usr/bin/perl
use strict;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my $uid = $<;
my @out;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover standby database until cancel;
auto
EOF2
`
};}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover standby database until cancel;
auto
EOF2
`;
}
print @out;

```

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 라이선스와 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 라이선스도 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 라이선스를 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 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

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