



Dell Storage Center와 Red Hat Enterprise Linux(RHEL) 7x 모범 사례

Daniel Tan, 제품 전문가
Dell Storage Applications Engineering

2015 년 5 월

개정

날짜	개정	설명	작성자
2014 년 7 월	1.0	초안	Daniel Tan
2015 년 5 월	1.1	Dell Storage SCv2000 에 대한 연결 소개	Daniel Tan

본 백서는 정보를 제공하기 위한 용도로만 제공되며 오탈자 및 기술적으로 부정확한 부분이 있을 수 있습니다. 백서의 내용은 일체의 명시적 또는 묵시적 보증 없이 있는 그대로 제공됩니다.

© 2014–2015 Dell Inc. 저작권 본사 소유.

Dell, Dell 로고 및 Dell 배지는 Dell Inc.의 상표입니다.

본 설명서에서 특정 회사의 표시나 제품 이름을 지칭하기 위해 기타 상표나 상호를 사용할 수도 있습니다.

Dell 은 타사 상표 및 상호에 대해 어떠한 소유권도 갖지 않습니다.



목차

개정	2
실행 요약	6
1 개요	7
1.1 RHEL 7x 의 새로운 기능	7
2 볼륨 관리	8
2.1 새 볼륨 스캔	8
2.2 파티션	9
2.3 LVM(Logical Volume Manager)	10
2.3.1 LVM 구성 및 SCSI UNMAP/TRIM	10
2.4 SCSI UNMAP/TRIM 및 파일 시스템	10
2.5 지속적인 장치 관리	11
2.5.1 파일 시스템 볼륨 레이블	11
2.5.2 범용 고유 식별자	12
2.5.3 /etc/fstab and /boot/grub2/grub.cfg 의 영구 명명	12
2.5.4 볼륨 레이블 및 UUID 스크립트	13
2.6 Replay View 볼륨 사용	14
2.7 볼륨 확장	16
2.8 볼륨 제거	17
2.9 SAN 에서 부팅	19
2.9.1 SAN Replay 볼륨에서 부트 캡처	23
2.9.2 SAN Replay View 볼륨에서 부팅 복구	26
3 서버 구성	27
3.1 Fibre Channel 및 modprobe	27
3.1.1 구성	27
3.1.2 modprobe 및 mkinitrd 다시 로드	27
3.2 iSCSI	28
3.2.1 구성	28
3.2.2 iSCSI 시간 초과	32
3.2.3 iSCSI 및 /etc/fstab	33



3.3	직렬 연결된 SCSI	34
3.3.1	SAS 드라이버	34
3.3.2	SAS /etc/multipath.conf	34
3.3.3	FC/iSCSI 및 SAS	35
3.3.4	Linux 에서 SAS 장치 식별	35
3.3.5	Dell SCv2000 의 SAS 장치 식별	36
3.3.6	구성된 다중 경로	38
3.3.7	SAS 큐 크기	39
3.3.8	SAS 에서 부팅	39
3.4	큐 크기 관리	39
3.5	/etc/multipath.conf 및 Storage Center 장치 정의	40
3.5.1	다중 경로 별칭	42
3.6	다중 경로 환경	42
3.6.1	다중 경로/SCSI 제한 시간 및 큐	43
3.6.2	PortDown 제한 시간	43
3.6.3	매개변수 확인	44
3.7	단일 경로 환경	44
3.7.1	PortDown 제한 시간	44
3.7.2	매개변수 확인	44
4	성능 고려 사항	45
4.1	조정 프로필	46
4.2	다중 볼륨	46
4.3	HBA 큐 크기	48
4.4	SCSI UNMAP/TRIM	48
4.5	Linux SCSI 장치 큐 변수	49
4.5.1	커널 I/O 스케줄러	49
4.5.2	read_ahead_kb	50
4.5.3	nr_requests	50
4.6	iSCSI	50
5	유용한 도구	51
5.1	lsscsi 명령	51



5.2	scsi_id 명령	52
5.3	dmsetup 명령	53
5.4	dmesg 명령어	54
5.5	/proc/scsi/scsi 파일	54
6	Dell Compellent Command Utility	55
6.1	Java 확인, CompCU 기능 구성 및 테스트	55
6.2	CompCU 를 사용하여 일반적인 작업 자동화	57
6.2.1	CompCU 를 사용하여 단일 볼륨 생성	57
6.2.2	CompCU 를 사용하여 Replay 및 Replay View 생성	58
6.2.3	CompCU 를 사용한 다중 볼륨의 신속 배포	58
A	추가 자료	59
B	Configuration details(구성 세부 정보)	60



실행 요약

Red Hat Enterprise Linux®(RHEL)는 매우 견고하고 확장 가능한 엔터프라이즈급 운영 체제입니다. RHEL 운영 체제는 본 백서에 제시된 모범 사례를 사용하여 올바르게 구성되었으므로 Dell™ Storage Center 에서 사용할 최적의 환경을 제공합니다. 이러한 모범 사례에는 볼륨 검색, 다중 경로, 파일 시스템 및 큐 크기 관리 구성에 대한 지침이 포함되어 있습니다.

이 백서의 범위에서는 Storage Center OS(SCOS) 버전 6.5.x 에서 6.6.x 까지의 RHEL 플랫폼 버전 7x 의 기능에 대해 설명합니다. 이러한 작업을 수행하기 위해 다양한 방법이 이미 마련되어 있으므로 이 백서는 최종 사용자와 시스템 관리자를 위한 참조 시작점을 제공합니다.

이 안내서는 거의 독점적으로 명령줄 인터페이스(CLI)에 초점을 맞추고 있습니다. 이는 UNIX 및 Linux 배포에서 가장 보편적으로 적용되기 때문입니다.



1 개요

Storage Center 는 업무상 중요한 데이터의 할당, 관리, 사용 및 보호의 복잡성을 제거하는 Linux 호환 및 SCSI-3 준수 디스크 볼륨을 제공합니다. 적절하게 구성된 Storage Center 를 사용하면 복잡한 RAID 구성 모범 사례와 함께 복잡한 물리적 디스크 구성 연습 및 관리가 필요하지 않습니다. 또한 Storage Center 는 저장소 계층에서 RAID 10 속도 및 안정성을 제공하므로 Linux 운영 체제 계층에서 볼륨이 추가로 RAID 를 관리할 필요가 없습니다.

미러링, 백업, 다중 파일 시스템, 다중 경로, SAN 에서 부팅 및 재해 복구와 같은 다양한 Linux 유틸리티를 Storage Center 볼륨과 함께 사용할 수 있습니다.

1.1 RHEL 7x 의 새로운 기능

RHEL 7x 는 신뢰성, 성능 및 확장성을 크게 향상시킵니다. 이 백서에서는 Dell Storage Center 제품 및 기능과의 통합을 지원하고 사용을 더욱 단순화하는 새로운 RHEL 7x 기능에 대해 설명합니다.

- XFS 는 부팅, 루트 및 사용자 데이터 파티션을 위한 새로운 기본 파일 시스템입니다.
- XFS 파일 시스템 크기 제한이 100TB 에서 500TB 로 증가했습니다. ext4 파일 시스템 크기 제한은 16TB 에서 50TB 로 증가했습니다.
- iSCSI 및 FCoE 대상의 새로운 소프트웨어 구현은 사용자 공간 대신 커널에 있습니다.



2 볼륨 관리

Linux 에서 볼륨을 관리하는 방법을 이해하려면 /sys 의 사 파일 시스템에 대한 기본적인 이해가 필요합니다. /sys 파일 시스템은 커널 및 모듈의 다양한 요소와 상호 작용할 수 있는 파일 구조입니다. 읽기 전용 파일은 현재 값을 저장하지만 쓰기/쓰기 파일은 올바른 명령으로 이벤트를 트리거합니다. 일반적으로 cat 및 echo 명령은 리디렉션과 함께 STDIN 절로 사용되어 기존의 텍스트 편집기에서 열립니다.

HBA(FC, iSCSI 및 SAS)와 상호 작용하기 위해 명령은 /sys/class/scsi_host/ 폴더에 있는 특수 파일에 대해 실행됩니다. 다중 포트 카드의 각 포트는 고유한 HBA 를 나타내며 각 HBA 에는 스캔을 실행하고 HBA 매개변수를 읽는 파일이 포함된 자체 hostX 폴더가 있습니다. 이 폴더 레이아웃, 파일 및 기능은 HBA 공급업체 또는 유형(예: QLogic Fibre Channel, Emulex Fibre Channel, 소프트웨어-iSCSI 기반 HBA 또는 Dell 12Gbps SAS HBA)에 따라 달라질 수 있습니다.

2.1 새 볼륨 스캔

QLogic 24xx/25xx 시리즈 HBA 및 Emulex HBA 에 필요한 드라이버 모듈은 기본 커널 코드에 포함되어 있습니다. 다음 지침은 기본 HBA 드라이버 모듈에 적용됩니다. 공급업체(QLogic, Emulex) 독점 드라이버가 사용된 경우 지시 사항 및 자세한 내용은 공급업체 관련 설명서를 참조하십시오.

Linux 호스트는 즉시 LUN ID 0 을 검색할 수 없습니다. LUN ID 0 은 부팅 중에만 검색할 수 있으며 SAN 환경에서 부팅할 때 운영 체제 볼륨용으로 예약되어 있습니다. 다른 모든 볼륨은 LUN ID 1 이상과 연관되어야 합니다.

다음 샘플 스크립트는 명시된 보증이나 어떤 종류의 지원 없이 있는 그대로 제공됩니다. 이 스크립트는 Linux 운영 체제의 주요 개정 번호를 식별하고 /sys/class/scsi_host/ 폴더 내의 각 hostX 장치에 echo 명령을 적용합니다. 이 스크립트는 HBA 포트를 스캔하여 저장소 어레이에서 호스트에 제공되는 기존 및 새 볼륨을 검색하고 식별합니다. 이 스크립트는 호스트에 제공된 FC 및 iSCSI 장치를 모두 검색하는 데 사용할 수 있으며 Red Hat Linux 버전 5x 에서 7x 에 적용됩니다.

STDOUT 는 이 스크립트에서 생성되지 않습니다. /var/log/messages 의 내용이나 dmesg 또는 lsscsi 명령의 출력을 확인하여 새로 검색된 볼륨을 식별하십시오.

새 볼륨을 매핑하고 검색하는 동안 HBA 를 다시 스캔해도 호스트에 부정적인 영향을 미치지 않습니다.

```
#!/bin/bash
```

```
OSMajor=`uname -r | awk -F. '{print $(NF-1)}'`
```

```
echo "INFO: OS Major rev. ${OSMajor} detected!"
```

```
if [ "${OSMajor}" = "el7" -o "${OSMajor}" = "el6" ]; then
for i in /sys/class/scsi_host/*
do
```



```

        echo "- - -" >> ${i}/scan
    done
elif [ "$(uname -r | awk -F. '{print $(NF)}')" = "el5" ]; then
    echo "INFO: OS Major rev. el5 detected instead!"
    for i in /sys/class/scsi_host/*
    do
        echo 1 >> ${i}/issue_lip
        echo "- - -" >> ${i}/scan
    done
else
    echo "WARN: OSMajor parameter of unknown value, exit 1"
    exit 1
fi

```

이 스크립트의 샘플 출력:

```

# ./scan_bus.sh
INFO: OS Major Rev. el7 detected!

```

또는 sg3_utils 패키지를 설치하면 /usr/bin 폴더에 있는 기본 Red Hat 명령 rescan-scsi-bus.sh 가 제공됩니다.

```

# /usr/bin/rescan-scsi-bus.sh
Scanning SCSI subsystem for new devices
[snip]
0 new or changed device(s) found.
0 remapped or resized device(s) found.
0 device(s) removed.

```

2.2 파티션

부팅 볼륨이 아닌 볼륨에는 파티션(및 파티션 테이블)이 필요하지 않습니다. Storage Center 의 프로비저닝된 볼륨을 전체 드라이브로 사용하는 것이 좋습니다. 이 경우 볼륨이 프로비저닝된 계층의 모든 디스크에서 볼륨을 스트라이핑하는 Storage Center 의 고유한 장점을 활용합니다. RHEL 7x 및 기본 설치를 사용하면 부팅 볼륨이 두 가지로 분할됩니다. XFS 파일 시스템(RHEL 7x 의 기본값)은 부팅 파티션에 적용되고 다른 파티션은 LVM(Logical Volume Manager)에 의해 관리됩니다.

```

# parted
GNU Parted 3.1
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) select /dev/sda
Using /dev/sda
(parted) print
Model: SEAGATE ST9146803SS (scsi)
Disk /dev/sda: 147GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

```



Disk Flags:

Number	Start	End	Size	Type	File system	Flags
1	1049kB	525MB	524MB	primary	xfs	boot
2	525MB	147GB	146GB	primary		lvm

2.3 LVM(Logical Volume Manager)

LVM(Logical Volume Manager)을 적용하고 사용하여 Linux 에서 볼륨을 관리할 수 있습니다. 볼륨에 LVM 메타데이터(예: LVM 서명)를 설치하고 그에 따라 물리적 볼륨(PV), 논리 볼륨(LV) 및 볼륨 그룹(VG)을 고유하게 식별합니다. LVM 을 사용하여 Storage Center 볼륨(vgexport, vgexport 및 기타 LVM 관리 명령 사용 포함)을 보다 포괄하고 관리하는 것은 이 백서의 범위를 벗어납니다.

Storage Center 볼륨을 전체 드라이브로 사용하는 것이 가장 좋습니다. 그러나 LVM 은 저장소 계층에서 제공하지 않는 기능이나 이점을 제공할 수 있는 경우 계속 사용할 수 있습니다.

2.3.1 LVM 구성 및 SCSI UNMAP/TRIM

LVM 을 Storage Center 볼륨과 함께 사용하면 LVM 을 SCSI UNMAP/TRIM 명령을 준수하도록 구성하고 이러한 명령을 다시 Storage Center 에 전달할 수 있습니다.

이 시나리오는 논리 볼륨이 구성원 볼륨 그룹에서 제거되는 경우에 적용됩니다. 이 제거된 논리 볼륨에서 복구된 공간은 해제되어 페이지 풀로 반환된 Storage Center 페이지와 관련됩니다.

LVM 구성은 /etc/lvm/lvm.conf 파일을 편집하여 수행합니다.

1. /etc/lvm/lvm.conf 파일을 편집하십시오.
2. 키 값 쌍 "issue_discards = 0"을 "issue_discards = 1"로 변경하십시오.
3. 변경 사항을 저장하고 편집기 세션을 종료하십시오.

2.4 SCSI UNMAP/TRIM 및 파일 시스템

SCSI UNMAP/TRIM 의 동작은 파일 시스템 계층(예: ext4, xfs 또는 btrfs)에서 작동하도록 구성할 수도 있습니다.

이 시나리오는 ext4, xfs 또는 btrfs 파일 시스템에서 파일과 디렉터리가 제거되는 경우에 적용됩니다. 이러한 제거된 파일 및 디렉터리에서 복구된 공간은 해제되어 페이지 풀로 반환된 Storage Center 페이지와 관련됩니다.

파일 시스템에서 SCSI UNMAP/TRIM 기능을 사용 가능하게 하려면 mount 명령을 사용하십시오. 이 mount 명령 매개변수는 파일 시스템 유형(ext4, xfs 또는 btrfs)에 관계없이 비슷한 방식으로 적용됩니다.

```
# mount -o discard /dev/mapper/<volume_name> /<mountpoint_name>
```

이 마운트 매개변수는 파일 시스템의 /etc/fstab 파일에 적절한 플래그를 추가하여 다시 부팅할 때도 영구적으로 만들 수 있습니다.



```
# cat /etc/fstab
[snip]
/dev/mapper/VolGroup00-home /home          xfs     defaults          1 2
/dev/mapper/VolGroup00-swap swap      swap     defaults          0 0
/dev/mapper/<volume_name> /<mountpoint_name> xfs     defaults,discard 0 0
[snip]
```

2.5 지속적인 장치 관리

Linux 에서 검색된 볼륨에는 서버를 SAN 에 연결하는 HBA 포트에 사용되는 Linux 검색 방법에 따라 /dev/sdd 및 /dev/sdf 와 같은 장치 지정이 제공됩니다.

다른 용도로 사용되는 이 /dev/sdX 장치 이름은 /etc/fstab 파일의 항목을 포함하여 마운트 명령의 볼륨을 지정합니다. 정적 디스크 환경에서 /dev/sdX 장치 이름은 /etc/fstab 파일의 항목에 적합합니다. 그러나 Fibre Channel 또는 iSCSI 연결의 동적 특성으로 인해 재부팅 시 Linux 가 이러한 디스크 지정을 지속적으로 추적하지 못합니다.

이러한 볼륨이 지속적 명명 스키마에 의해 지정되고 참조되도록 하는 여러 가지 방법이 있습니다. 이 섹션에서는 볼륨 레이블 또는 범용 고유 식별자(UUID)를 /dev/sdX 참조 볼륨과 함께 사용하는 방법에 대해 설명합니다. 볼륨 레이블은 Storage Center Replay 복구를 스크립팅할 때 매우 유용합니다. 예를 들어, 프로덕션 Storage Center Replay 의 Replay View 볼륨을 백업 서버에 마운트하는 작업이 포함됩니다. 이 경우 Replay View 볼륨은 관련 /dev/sdX 장치를 명시적으로 식별할 필요없이 해당 볼륨 레이블에 의해 참조될 수 있습니다. 볼륨 레이블은 볼륨에 저장된 메타데이터이며 Storage Center Replay 에서 잘라낸 Replay View 볼륨에 상속됩니다.

볼륨 레이블 또는 UUID 는 다중 경로 환경에서도 사용될 수 있습니다. 즉, 다중 경로 /dev/mapper/mpathX 장치 이름(또는 다중 경로 별칭, 3.5.1 "[다중 경로 별칭](#)" 섹션 참조)은 기본적으로 지속되며 다시 부팅할 때마다 변경되지 않습니다. 볼륨 레이블, UUID 또는 다중 경로 장치 이름은 로컬 Linux 호스트의 /etc/fstab 파일에 있는 항목에 대해 서로 바껴서 사용할 수 있습니다.

Replay View 볼륨이 대체 Linux 호스트로 볼륨을 복구 또는 재배치하는 경우 볼륨 레이블 또는 UUID 를 사용하는 것이 좋습니다. 이 값은 볼륨을 고유하게 식별하기 때문이며 대조적으로 다중 경로 이름은 다를 수 있습니다(/etc/multipath.conf 파일의 구성에 따라 다름).

2.5.1 파일 시스템 볼륨 레이블

볼륨에 파일 시스템을 만들 때 또는 나중에 다른 명령을 사용하여 파일 시스템 볼륨 레이블을 적용할 수 있습니다. 다른 파일 시스템(예: ext4 또는 xfs)은 파일 시스템 메타데이터 스키마가 다르며 다른 명령을 사용하여 이 데이터를 보고 관리하고 변경할 수 있습니다.

파일 시스템 볼륨 레이블은 /dev/disk/by-label/ 폴더에 작성됩니다. 이 폴더의 항목은 /dev 폴더의 각 장치에 대한 기호 링크로 작성됩니다. /dev/disk/by-label/ 폴더는 동적으로 관리되며 Linux 호스트의 어떤 볼륨에도 볼륨 레이블이 적용되지 않으면 존재하지 않습니다.



아래의 예는 이러한 개념 중 일부를 보여줍니다. 2.5.4 "[볼륨 레이블 및 UUID 스크립트](#)" 섹션의 샘플 스크립트는 모든 Linux 다중 경로 장치 이름을 구문 분석하고 다중 경로 UUID, 알려진 파일 시스템 유형, 파일 시스템 볼륨 레이블 및 파일 시스템 UUID 값을 STDOUT 로 표시합니다. 이 스크립트는 명시된 보증이나 지원 없이 있는 그대로 제공됩니다.

- 파일 시스템과 레이블을 동시에 적용하려면:

```
# mkfs.ext4 -L My_ext4_vol /dev/sdX
# mkfs.xfs -L My_xfs_vol /dev/mapper/mpathX
```

- 기존 파일 시스템에 볼륨 레이블을 적용하려면(다른 파일 시스템 유형에 사용되는 다른 명령 기록):

```
# tune2fs -L My_ext4_vol /dev/sdX
# xfs_admin -L My_xfs_vol /dev/mapper/mpathX
```

- 기존 파일 시스템에서 볼륨 레이블을 제거하려면(다른 파일 시스템 유형에 사용되는 다른 명령 기록):

```
# tune2fs -L "" /dev/sdX
# xfs_admin -L "--" /dev/mapper/mpathX
```

2.5.2 범용 고유 식별자

UUID 값은 존재하며 볼륨 메타데이터의 여러 계층에서 지속됩니다. Linux 의 각 다중 경로 장치를 고유하게 식별하고 관리하기 위해 UUID 는 device-mapper 드라이버에 의해 생성되고 사용됩니다. UUID 는 LVM 메타데이터 내에서 생성됩니다(LVM 에서 볼륨을 관리하는 경우). UUID 는 또한 파일 시스템 계층 내에서 생성 및 관리되어 각 파일 시스템 볼륨을 고유하게 식별합니다. UUID 는 다중 경로 정의, LVM 또는 파일 시스템 생성 중에 동적으로 생성됩니다. UUID 는 다른 레이어와 독립적으로 각 레이어 내에서 변경할 수 있지만 의도가 명확하게 정의된 경우가 아니면 권장하지 않습니다.

모든 UUID 항목은 /dev/disk/by-uuid/ 폴더에 만들어집니다. 이 폴더의 항목은 /dev 폴더의 각 장치에 대한 기호 링크로 작성됩니다.

2.5.4 "[볼륨 레이블 및 UUID 스크립트](#)" 섹션의 샘플 스크립트는 모든 Linux 다중 경로 장치 이름을 구문 분석하고 다중 경로 UUID, 알려진 파일 시스템 유형, 파일 시스템 볼륨 레이블 및 파일 시스템 UUID 값을 STDOUT 로 표시합니다. 이 스크립트는 명시된 보증이나 지원 없이 있는 그대로 제공됩니다.

2.5.3 /etc/fstab and /boot/grub2/grub.cfg 의 영구 명명

LABEL= 또는 UUID= 구문은 마운트 명령, /etc/fstab 파일의 항목 및 /boot/grub2/grub.cfg 파일 또는 스왑 파티션을 포함하여 다양한 위치에서 볼륨을 참조하는 데 사용될 수 있습니다. 이는 검색 장치 이름 지정, 다중 경로 구성 또는 사전 정의된 다중 경로 별명에 관계없이 볼륨을 고유하게 식별할 수 있는 자유를 제공합니다.

아래의 샘플 /etc/fstab 출력은 이러한 개념 중 일부를 사용하는 방법을 보여줍니다.

```
# cat /etc/fstab
[snip]
/dev/mapper/VolGroup00-home /home          xfs     defaults                1 2
UUID=8284393c-18aa-46ff-9dc4-0357a5ef742d    swap     swap defaults 0 0
LABEL=TestVol /vol_001    xfs     defaults,discard 0 0
[snip]
```



아래의 샘플 /etc/grub2.conf 출력은 이러한 개념 중 일부를 사용하는 방법을 보여줍니다.

```
# cat /boot/grub2/grub.cfg
[snip]
linux16 /vmlinuz-3.10.0-123.el7.x86_64 root=UUID=35bea1c0-ce32-42a4-8e36-
72fd5e77471d ...
[snip]
```

2.5.4 볼륨 레이블 및 UUID 스크립트

이 샘플 스크립트는 vendor = "COMPELNT"로 식별된 모든 장치를 구문 분석하고 파일 시스템이 볼륨에 적용되었는지 확인한 다음, 발견된 디스크 레이블 값, mpath-layer UUID 및 disk-layer UUID 값을 추출하여 표시합니다. 이 스크립트는 명시된 보증이나 어떠한 종류의 지원 없이 있는 그대로 제공됩니다.

```
#!/bin/bash

MP=/usr/sbin/multipath
BLKID=/usr/sbin/blkid

for i in `${MP} -ll | grep COMPELNT | cut -d" " -f1 | sort -k1`
do
    echo "INFO: /dev/mapper/${i}"
    FsTyp=`${BLKID} /dev/mapper/${i} | awk -F" " '{print $(NF)}' | cut -d= -
f2 | cut -d\" -f2`
    if [ "${FsTyp}" = "" ]; then
        echo "WARN: No filesystem detected"
    else
        echo "Multipath Info:"
        /usr/sbin/dmsetup info /dev/mapper/${i} | grep UUID
        echo "Filesystem Details:"
        echo "Type: ${FsTyp}"
        case ${FsTyp} in
            xfs)
                Cmd="/usr/sbin/xfs_admin -lu"
                ${Cmd} /dev/mapper/${i}
                ;;
            ext4|ext3)
                Cmd="/usr/sbin/tune2fs -l"
                ${Cmd} /dev/mapper/${i} | egrep 'volume|UUID'
                ;;
            *)
                echo "WARN: Filesystem unknown"
                ;;
        esac
    fi
done
```



이 스크립트의 샘플 출력은 다음과 같습니다.

```
# ./get_UUID.sh
[snip]
INFO: /dev/mapper/vol_001
Multipath Info:
UUID: mpath-36000d31000006500000000000000017f2
Filesystem Details:
Type: xfs
label = "TestVol"
UUID = fbf5bfb-94b9-4827-a103-516045b9b608

INFO: /dev/mapper/vol_002
Multipath Info:
UUID: mpath-36000d31000006500000000000000017f3
Filesystem Details:
Type: LVM2_member
WARN: Filesystem unknown
[snip]
```

2.6 Replay View 볼륨 사용

XFS 파일 시스템을 사용하면 Storage Center Replay 및 Replay View 볼륨과 간단하고 효과적으로 통합할 수 있습니다. 볼륨의 맨 위에 있는 적용된 XFS 파일 시스템은 볼륨 UUID 를 포함하여 볼륨의 XFS 메타데이터를 생성, 유지 및 관리합니다. 이 볼륨 UUID 는 연결된 호스트에 관계없이 Linux 가 이 볼륨을 고유하게 식별하도록 지원합니다.

이 방법은 모든 Linux 호스트의 XFS 파일 시스템 관리에 적용할 수 있습니다. Replay View 볼륨이 대체 Linux 호스트(UUID 포함)에 제공되면 다음 방법 중 하나를 사용하지 않고도 볼륨이 정상적으로 마운트됩니다.

알려진 XFS 파일 시스템의 볼륨 레이블 및 UUID 는 아래 명령으로 표시할 수 있습니다.

```
# xfs_admin -ul /dev/mapper/vol_001
UUID = fbf5bfb-94b9-4827-a103-516045b9b608
label = "TestVol"
```

파일 시스템의 이 UUID 는 원래 Storage Center 볼륨에서 작성된 Replay 또는 Replay View 볼륨에 의해 상속됩니다. 따라서 다중 경로 장치가 고유한 ID 를 갖고 있더라도 모든 Replay View 볼륨을 원래 호스트에 자동으로 표시하고 마운트할 수 없습니다. 아래 예에서 Replay View 볼륨은 XFS 파일 시스템 기반 /dev/mapper/vol_001 장치에서 만들어지고 /vol_001 로 마운트됩니다. 이 Replay View 볼륨은 /etc/multipath.conf 파일을 사용하여 /dev/mapper/vol_004 로 검색, 식별 및 별칭 설정됩니다.

참고: /dev/mapper/vol_004 를 /vol_004 로 마운트하려는 시도는 UUID 가 고유하지 않고 이미 Linux 호스트에서 사용 중이기 때문에 실패합니다.



```
# df -k
Filesystem                1K-blocks      Used Available Use% Mounted on
[snip]
rhevsn:/nfsroot            103213056 54833152  43137024   56% /Tools
/dev/mapper/vol_001        52418560  1939052  50479508    4% /vol_001
[snip]
```

/dev/mapper/vol_004 를 /vol_004 에 마운트하려고 시도하면 아래 오류가 반환됩니다.

```
# mount -o discard, sync /dev/mapper/vol_004 /vol_004
mount: wrong fs type, bad option, bad superblock on /dev/mapper/vol_004,
       missing codepage or helper program, or other error
```

In some cases useful info is found in syslog - try
dmesg | tail or so.

nouuid 매개변수와 함께 /dev/mapper/vol_004 를 /vol_004 에 마운트하려고 하면 볼륨을 /vol_004 에 성공적으로 마운트 할 수 있습니다.

```
# mount -o discard, sync, nouuid /dev/mapper/vol_004 /vol_004; df -k
```

```
# df -k
Filesystem                1K-blocks      Used Available Use% Mounted on
[snip]
rhevsn:/nfsroot            103213056 54833152  43137024   56% /Tools
/dev/mapper/vol_001        52418560  1939052  50479508    4% /vol_001
/dev/mapper/vol_004        52418560  1939052  50479508    4% /vol_004
[snip]
```

또는 /dev/mapper/vol_004 와 연관된 UUID 값을 영구히 변경하여 마운트 명령과 함께 nouuid 매개변수를 지정하지 않아도 됩니다. 다음과 같이 이 작업을 수행할 수 있습니다(마운트 해제된 파일 시스템에 적용).

```
# xfs_admin -U generate /dev/mapper/vol_004
[[[ [[[[ [ UUID [[
[[ SB [[
new UUID = 26b4a32d-4a3a-405f-899c-7bb63087cc7b
```



2.7 볼륨 확장

RHEL 7x에서는 Linux의 볼륨 및 파일 시스템 확장이 간단하고 효율적입니다. 기본 Storage Center 볼륨의 크기가 증가된 후 볼륨 맨 위에 있는 파일 시스템은 기본 볼륨의 전체 한계로 크기를 조정하거나 시스템 관리자가 지정한대로 사전 결정된 크기로 조정할 수 있습니다. 볼륨 및 XFS 파일 시스템 확장 프로세스는 아래 단계에 설명되어 있습니다.

1. Storage Center 볼륨을 확장합니다.
 - a. Storage Center 볼륨을 마우스 오른쪽 단추로 클릭합니다.
 - b. **Expand Volume(볼륨 확장)**을 선택합니다.
 - c. 볼륨의 새 크기를 입력합니다.
 - d. **OK(확인)**를 클릭하여 확장을 실행합니다.
2. 다중 경로 장치의 각 경로에 대한 드라이브 지오메트리를 다시 스캔하고 `# echo 1 >> /sys/block/sdX/device/rescan` □□□□□□.
3. 다중 경로 맵의 크기를 조정하고 `# multipathd -k"resize map <devicename>"`을 실행하십시오.

참고:

- a. 여기서 다중 경로 장치는 `/dev/mapper/<devicename>`입니다.
 - b. `-k` 매개변수와 명령 문자열 사이에는 공백이 없습니다.
4. XFS 파일 시스템을 확장합니다(이 예에서는 기본 볼륨의 경계 제한 크기까지).
 - a. `# xfs_growfs -d /<mountpoint_name>`

어떤 환경에서도 볼륨 또는 해당 부속 파일 시스템의 크기를 즉시 줄이는 것은 권장되지 않습니다. 데이터 무결성에 대한 위험과 잠재적인 영향을 크게 줄이는 대체 방법은 백업 및 복구, 볼륨 크기를 줄인 새로운 Storage Center 볼륨 또는 미러를 미러링하고 볼륨을 분할하여 볼륨의 내용을 재배포하는 것입니다.

이 샘플 스크립트는 명령줄(`/dev/mapper/<devicename>` 또는 `<devicename>` 만)에서 입력을 수신하고 다중 경로 장치를 지원하는 다양한 `/dev/sdX` 장치 이름을 결정합니다. 그런 다음 지정된 다중 경로 장치의 각 경로의 볼륨 지오메트리를 다시 스캔하고 업데이트하기 위해 `/sys/block/sdX/device/rescan` 파일에 1을 에코 설정합니다. 이 스크립트는 명시된 보증이나 어떠한 종류의 지원 없이 있는 그대로 제공됩니다.

```
#!/bin/bash
```

```
DevMapperDevice=${1}
```

```
if [ "${DevMapperDevice}" = "" ]; then
    echo "ERROR: No /dev/mapper/<devicename> defined, Exiting"
    exit 1
fi
```

```
for i in `multipath -l ${DevMapperDevice} | egrep -v "${basename
${DevMapperDevice}}|^size|policy" | awk '{print $3}'`
```



```
do
    echo "INFO: Rescanning /dev/${i} device"
    echo 1 >> /sys/block/${i}/device/rescan
done

echo "INFO: Multipath device /dev/mapper/${(basename ${DevMapperDevice})}
rescanned successfully"
```

이 스크립트의 샘플 출력은 다음과 같습니다.

```
# ./rescan_volume.sh /dev/mapper/vol_001
INFO: Rescanning /dev/sdf device
INFO: Rescanning /dev/sdk device
INFO: Rescanning /dev/sdg device
INFO: Rescanning /dev/sdj device
INFO: Multipath device /dev/mapper/vol_001 rescanned successfully
```

2.8 볼륨 제거

Linux 는 제공되는 각 볼륨에 대한 정보를 저장합니다. Storage Center 에서 볼륨이 매핑되지 않은 상태인 경우에도 Linux 는 다음 재부팅 때까지 해당 볼륨에 대한 정보를 유지합니다. 다시 부팅하기 전에 동일한 LUN ID 를 사용하는 동일한 대상의 볼륨이 Linux 에 제공되면 해당 볼륨에 대한 이전 데이터가 다시 사용됩니다. 이로 인해 볼륨의 잘못된 정보 및 관리가 발생할 수 있으며 비즈니스 환경에서 데이터 무결성에 잠재적으로 영향을 미칠 수 있습니다.

볼륨이 더 이상 사용되지 않는 것으로 간주되면 Linux 에서 모든 볼륨 정보를 항상 마운트 해제하고 제거하고 삭제하는 것이 좋습니다. Linux 볼륨 메타데이터 관리는 볼륨 자체에 저장된 실제 데이터를 삭제하지 않습니다.

다중 경로 볼륨 제거 프로세스는 아래 단계에 설명되어 있습니다.

1. 마운트된 볼륨에 대한 I/O 를 중지합니다.
2. 볼륨을 마운트 해제합니다.
3. /etc/multipath.conf 를 편집하고 이 볼륨을 참조하는 구문을 제거합니다.
4. 다중 경로 데몬을 다시 로드합니다(systemctl restart multipathd.service).
5. 아래의 스크립트를 실행하여 다중 경로 보호 장치 파일을 제거하십시오.
6. Storage Center 에서 볼륨에 대한 모든 매핑을 제거하십시오.

이 샘플 스크립트는 명령줄(/dev/mapper/<devicename> 또는 <devicename> 만)에서 입력을 수신하고 다중 경로 장치를 지원하는 다양한 /dev/sdX 장치 이름을 결정합니다. 그런 다음, /sys/block/sdX/device/delete 파일에 적절하게 1 을 에코하고 Linux 에서 볼륨 메타데이터를 삭제합니다. 이 스크립트는 명시된 보증이나 어떠한 종류의 지원 없이 있는 그대로 제공됩니다.



```
#!/bin/bash

DevMapperDevice=${1}

if [ "${DevMapperDevice}" = "" ]; then
    echo "ERROR: No /dev/mapper/<devicename> defined, Exiting"
    exit 1
fi

for i in `multipath -l ${DevMapperDevice} | egrep -v "$(basename
${DevMapperDevice})|^size|policy" | awk '{print $3}'`
do
    echo "INFO: Removing /dev/${i} device"
    echo 1 > /sys/block/${i}/device/delete
done

echo "INFO: Multipath device /dev/mapper/${(basename ${DevMapperDevice})} removed
successfully"
```

이 스크립트의 샘플 출력은 다음과 같습니다.

```
# ./rm_volume.sh /dev/mapper/vol_001
INFO: Removing /dev/sdf device
INFO: Removing /dev/sdk device
INFO: Removing /dev/sdg device
INFO: Removing /dev/sdj device
INFO: Multipath device /dev/mapper/vol_001 removed successfully
```

다중 경로에서 관리하지 않는 단일 볼륨을 제거하려면 아래 단계를 수행하십시오.

1. 마운트된 볼륨에 대한 I/O 를 중지합니다.
2. 볼륨을 마운트 해제합니다.
3. 명령 프롬프트에서 # echo 1 > /sys/block/sdX/device/delete 를 실행합니다.
4. Storage Center 에서 볼륨에 대한 모든 매핑을 제거하십시오.

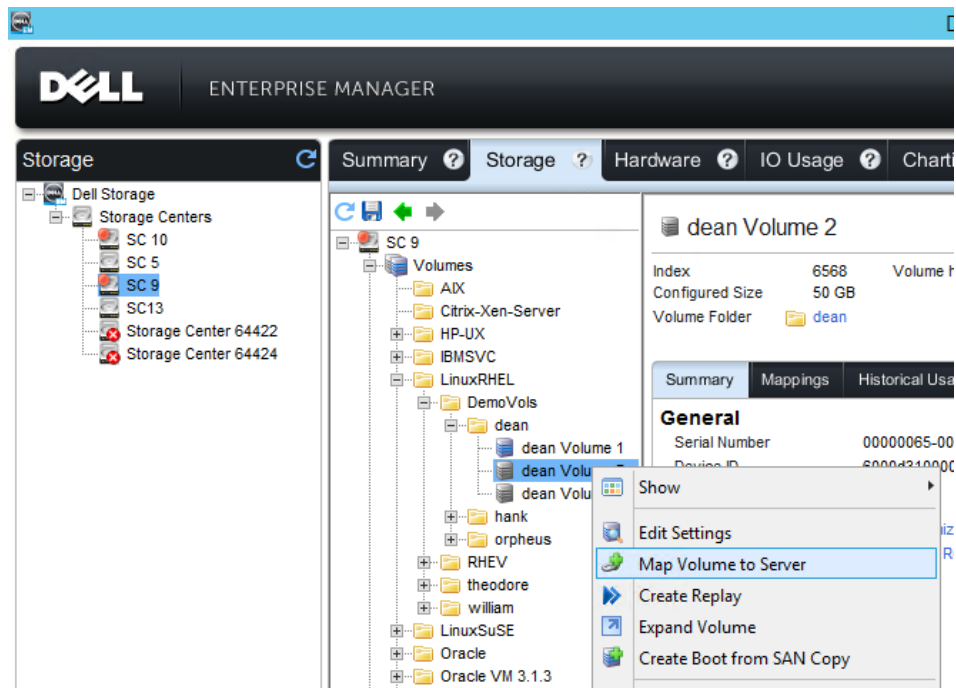


2.9 SAN 에서 부팅

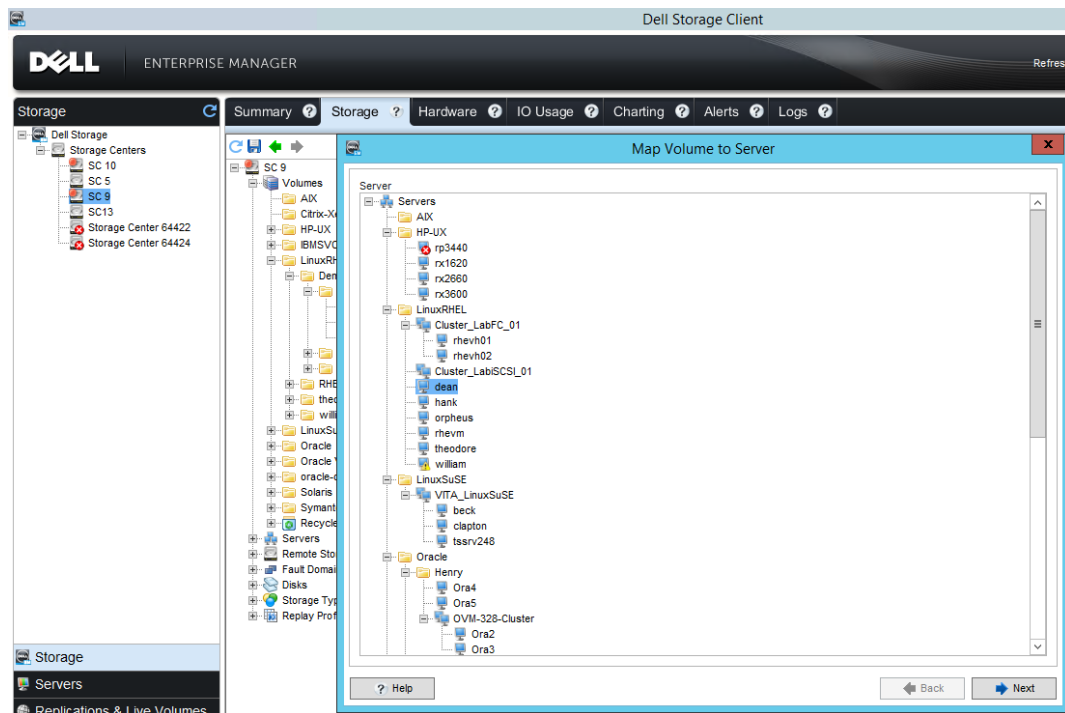
Linux 에서 Storage Center 볼륨을 부팅 가능한 볼륨으로 사용하면 시스템 및 저장소 관리자가 Storage Center Replay 및 Replay View 볼륨 기술의 이점을 보다 효과적으로 활용할 수 있습니다. Linux 부팅 볼륨의 Replay 에는 백업/복구 메커니즘 또는 업그레이드 이전 시점의 운영 체제 상태를 보존하는 두 가지 용도가 있습니다.

Storage Center 볼륨을 부팅 가능한 볼륨으로 사용하려면 대상 볼륨을 LUN ID 0 으로 대상 Linux 호스트에 제공해야 합니다. Storage Center 를 통한 볼륨 LUN ID 0 매핑은 아래와 같이 수행됩니다.

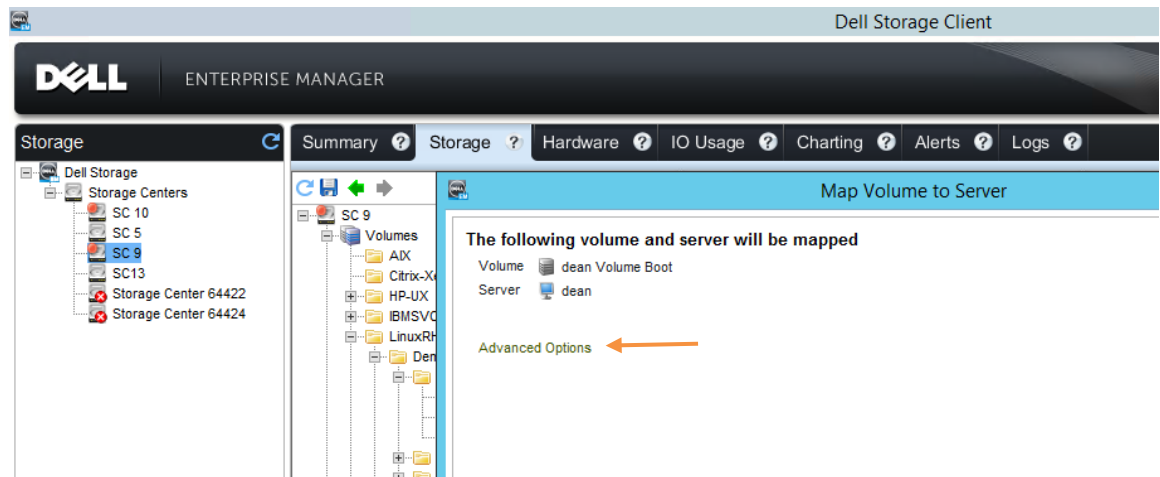
1. 볼륨을 마우스 오른쪽 단추로 클릭한 다음, **Map Volume to Server(서버에 볼륨 매핑)**을 선택합니다.



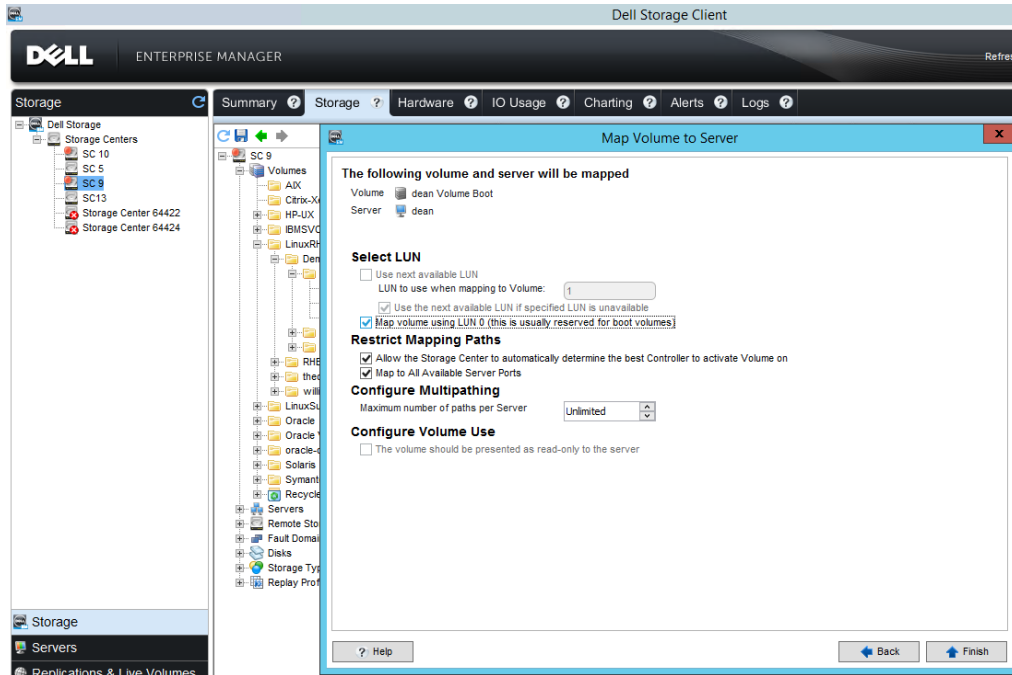
2. 볼륨을 매핑해야 하는 호스트를 식별하고, Next(다음)를 클릭합니다.



3. **Advanced Options(고급 옵션)**를 클릭합니다.



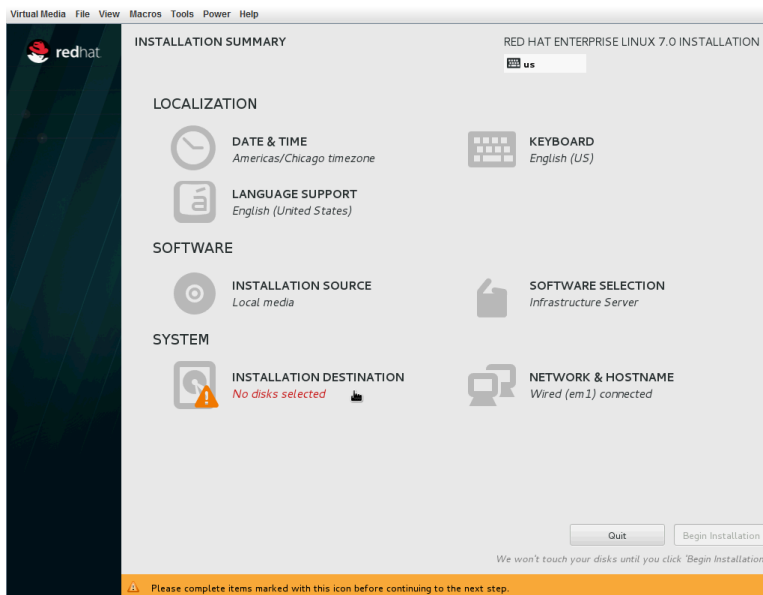
4. **Map volume using LUN 0**(LUN 0 을 사용하여 볼륨 매핑) 확인란 선택하고 **Finish**(마침)를 클릭하여 이 볼륨을 매핑합니다.



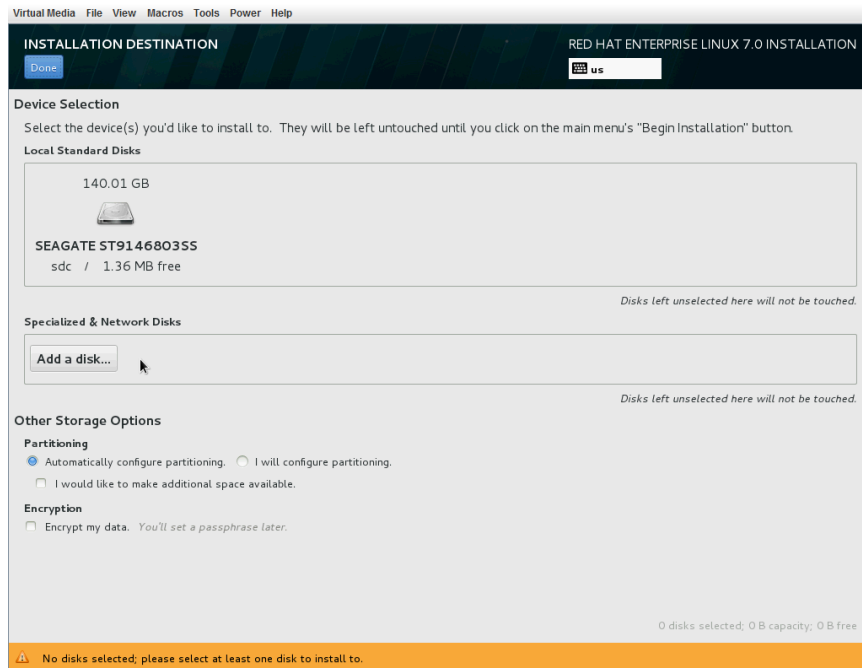
Linux 호스트 부팅 중에 HBA BIOS 부팅 프로세스가 중단되고 부팅 장치 순서에서 이 Storage Center 볼륨을 기본 부팅 장치로 식별하도록 지시합니다.

Linux 설치 프로세스 중에 이 Storage Center 볼륨을 식별하여 아래에 표시된 설치 대상으로 선택합니다.

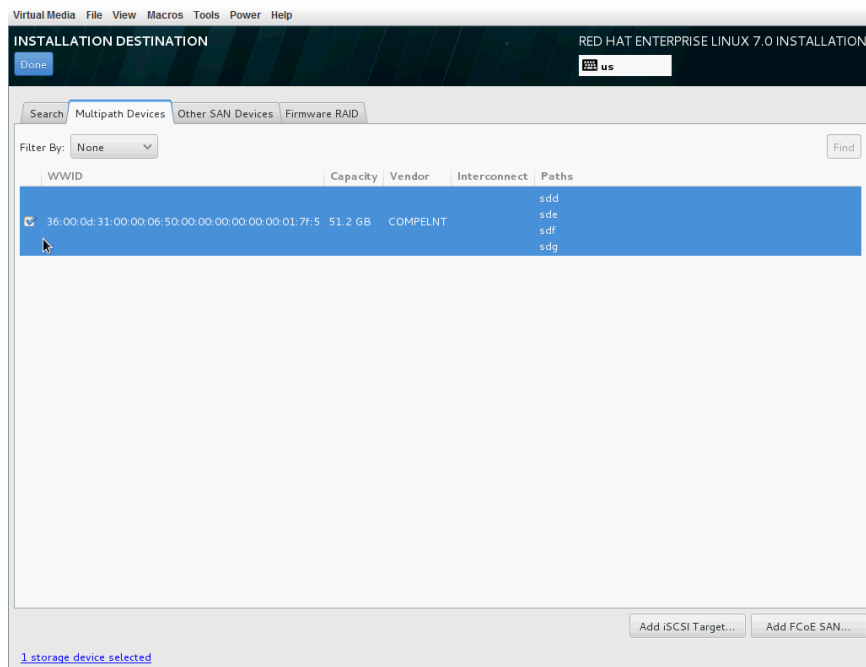
1. **Installation(설치)** 기본 메뉴에서 **Installation Destination(설치 대상)**을 클릭합니다.



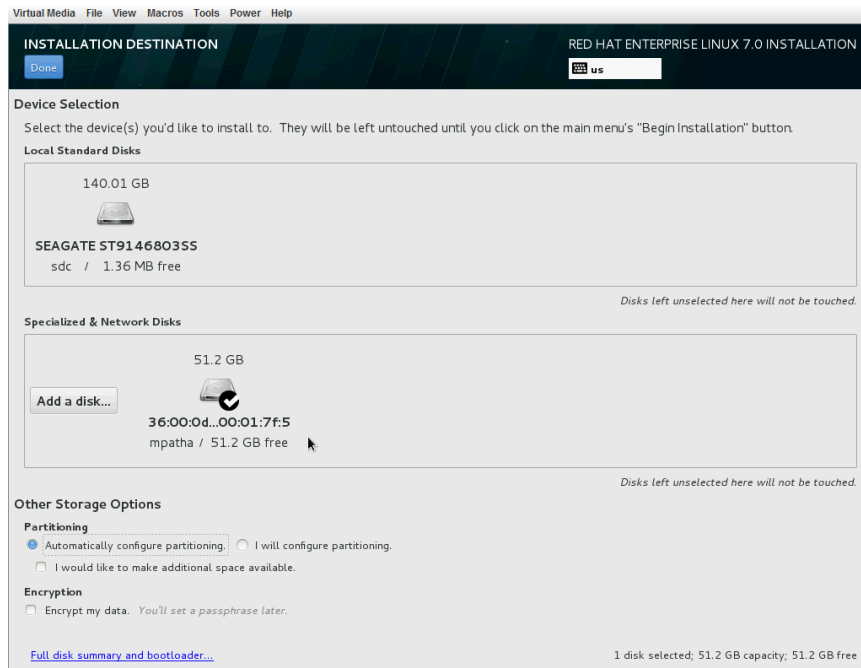
2. **Add a disk(디스크 추가)**를 클릭합니다.



3. **Multipath Devices(다중 경로 장치)** 탭에서 Storage Center 볼륨을 확인하고 그에 따라 확인란을 선택하십시오.



4. **Done(완료)**을 클릭합니다.



위에 표시된 Storage Center 볼륨은 Linux 운영 체제의 단일 설치 대상으로 식별 및 선택됩니다.

5. **Done(완료)**을 클릭하여 설치 기본 메뉴로 돌아갑니다.

2.9.1 SAN Replay 볼륨에서 부트 캡처

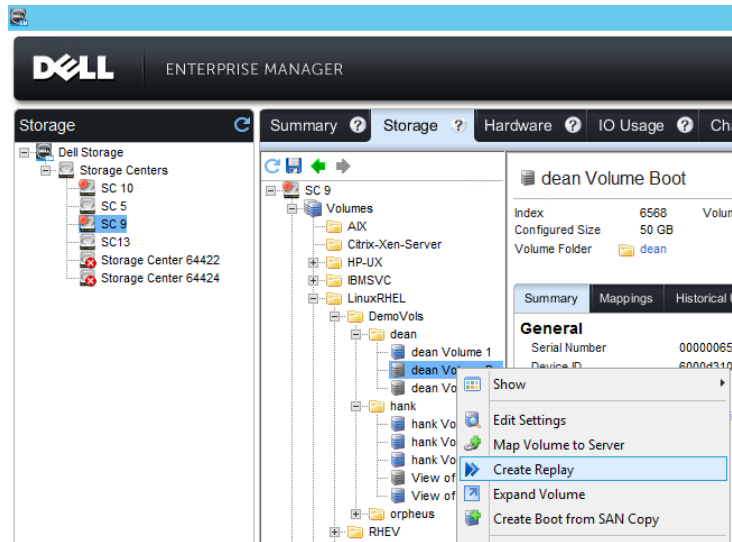
SAN 볼륨에서 부트의 Replay 볼륨을 캡처하는 것은 Storage Center 에서 간단합니다. 그러나 Replay 볼륨을 캡처하기 전에 I/O 를 중지하려면 Linux 호스트에서 LVM 이 아닌 모든 XFS 볼륨을 고정하는 것이 좋습니다.

이 시나리오(기본 Linux 설치 포함)에서 / 및 /boot 파일 시스템은 XFS 기반이며 다음 명령으로 고정되어 있습니다.

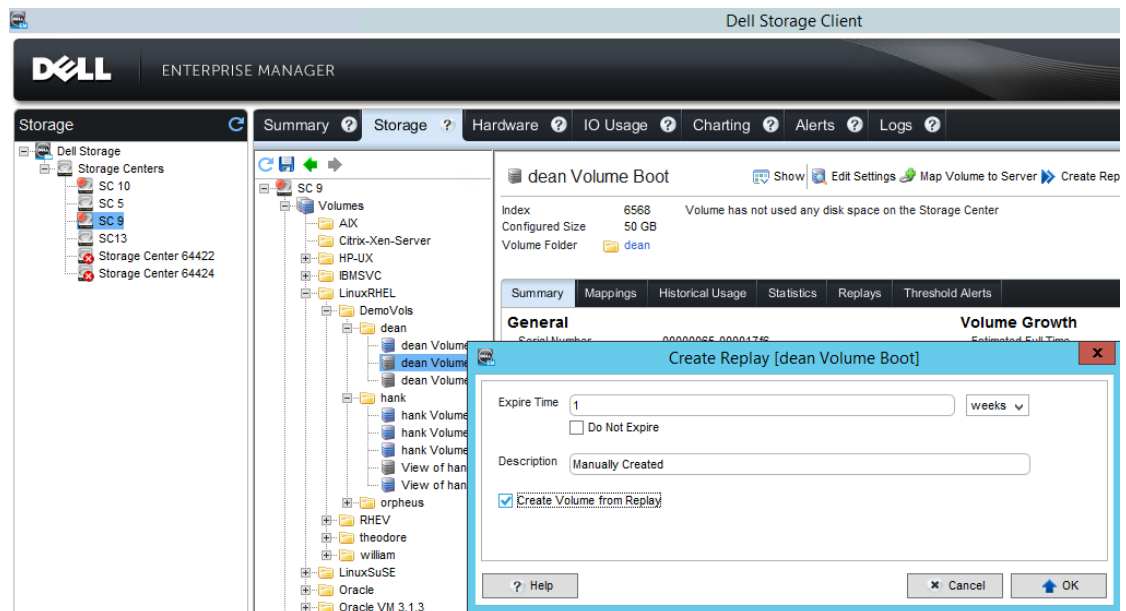
```
# xfs_freeze -f /  
# xfs_freeze -f /boot
```

Storage Center 를 통해 SAN 볼륨에서 부트의 Replay 볼륨을 캡처하는 것은 아래와 같이 수행됩니다.

1. 볼륨을 마우스 오른쪽 단추로 클릭한 다음 **Create Replay(Replay 생성)**를 선택합니다.



2. 볼륨에 1 주 만료 시간을 부여하고 **OK(확인)**를 클릭하여 Replay 를 만듭니다.

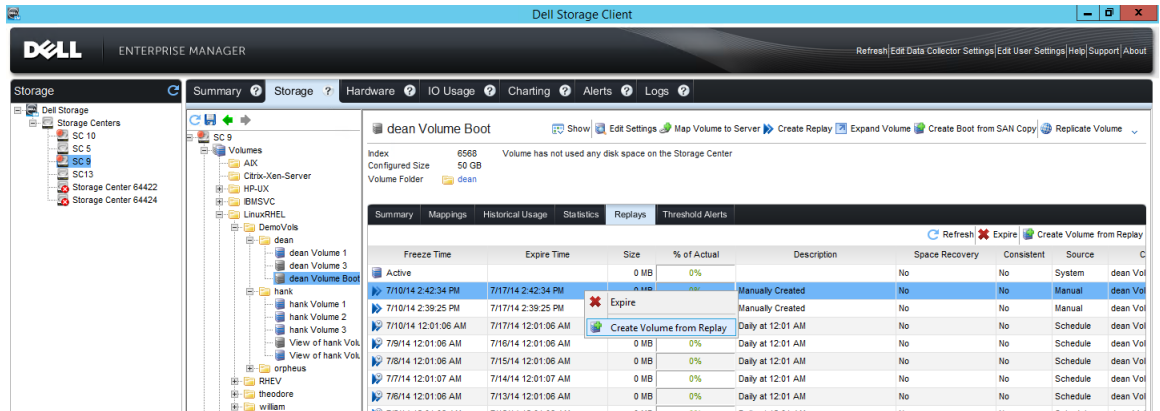


3. Replay 가 캡처된 후에는 / 및 /boot 파일 시스템이 XFS 기반이므로 다음 명령을 사용하여 해제해야 합니다.

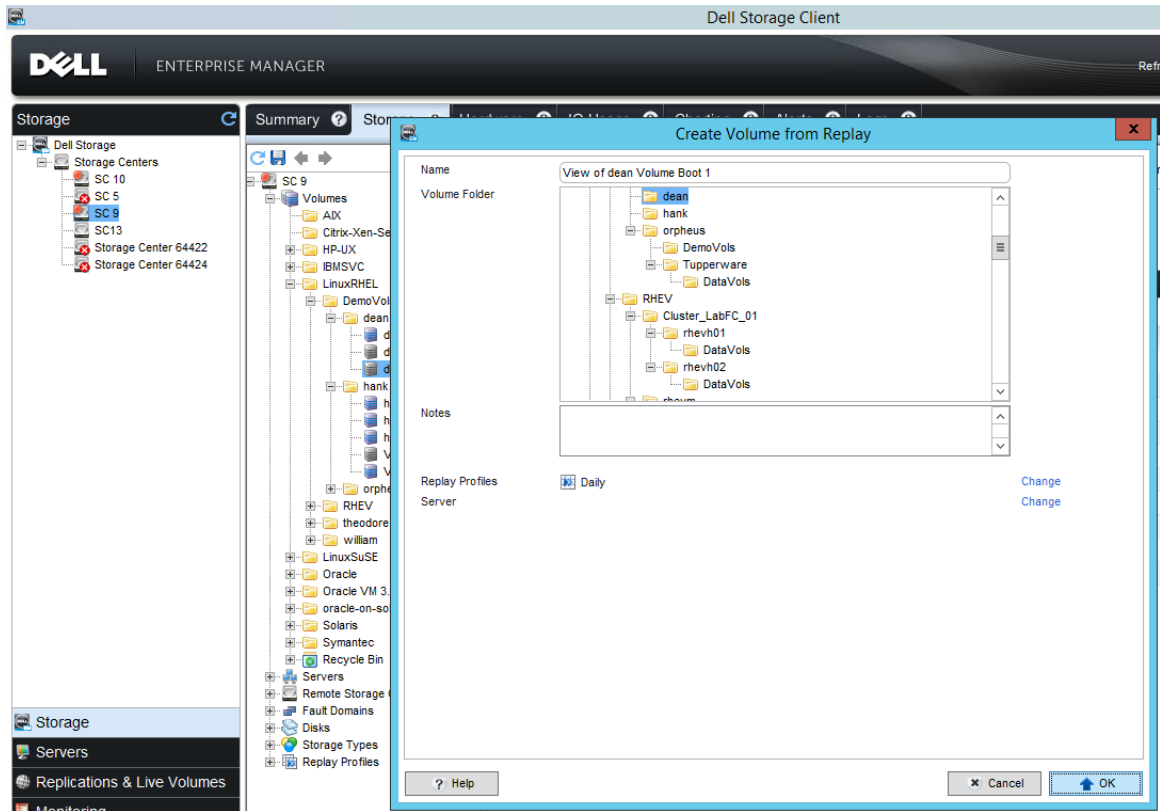
```
# xfs_freeze -u /
# xfs_freeze -u /boot
```

이 캡처된 Replay 의 Replay View 볼륨이 4 단계와 같이 생성됩니다.

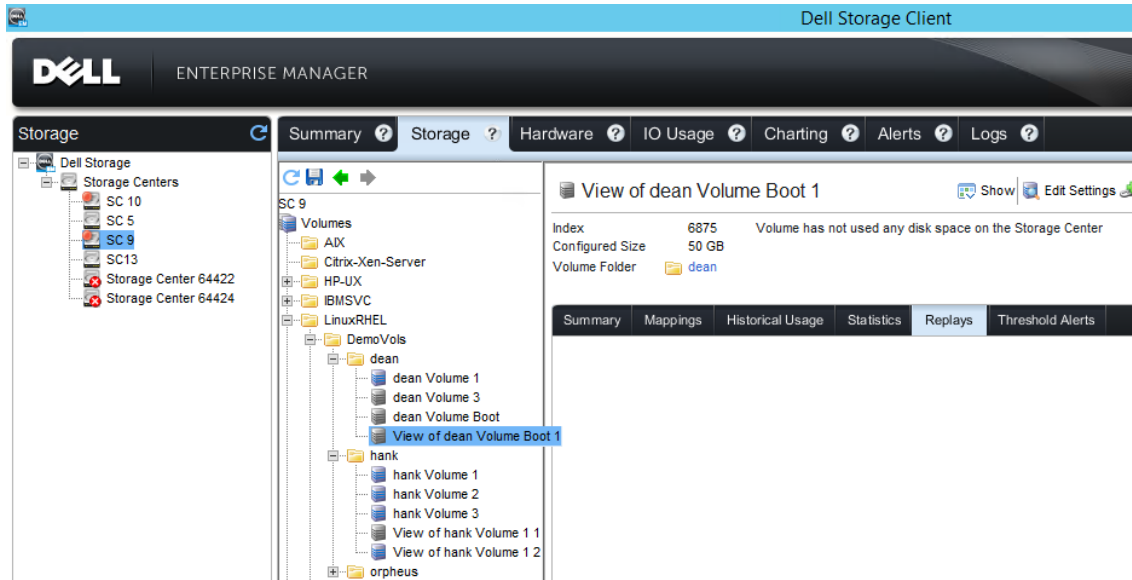
- 볼륨 **Replay** 탭에서 원하는 Replay 를 마우스 오른쪽 단추로 클릭한 다음, **Create Volume from Replay(Replay 에서 볼륨 생성)**를 클릭합니다.



- 이 Replay View 볼륨을 적절한 서버 호스트 개체에 연결하고 **OK(확인)**를 클릭합니다.



Replay View 볼륨은 자신이 생성된 볼륨 아래의 **Volumes(볼륨)** 트리에 표시됩니다.



2.9.2 SAN Replay View 볼륨에서 부팅 복구

동일한 호스트에 다시 제공될 때 SAN 볼륨에서 부트의 Replay View 볼륨이 자동으로 부팅 가능합니다. 시스템 장치 파일 이름 대신 `/etc/fstab`, `/boot/grub2/grub.cfg` 내부의 고유하고 식별 가능한 UUID 값을 통해 볼륨이 추적되기 때문입니다. 또한 `/etc/multipath/bindings` 및 `/etc/multipath/wwids` 내의 항목과 별칭도 부팅 장치 변경 사항을 반영하도록 자동으로 업데이트됩니다.

교대로 매핑된 Replay View 볼륨에서 부팅을 시도할 때 부팅 시간 충돌을 방지하기 위해 SAN 볼륨의 원래 부팅이 더 이상 서버 개체에 매핑되지 않도록 하십시오. Replay View 볼륨은 LUN ID 0 으로 호스트에 매핑되어야 합니다.

3 서버 구성

이 섹션에서는 Linux 호스트의 I/O 스택의 다양한 구성 측면에 대해 설명합니다. Linux I/O 스택은 Linux 호스트가 작동하는 환경의 필요에 맞게 정확하게 구성될 수 있습니다 구성 측면에는 HBA 포트 재시도 횟수, 큐 크기, SCSI 및 iSCSI 장치 제한 시간 값 조정 등이 포함됩니다

3.1 Fibre Channel 및 modprobe

Linux 호스트의 다른 측면 중 Linux modprobe 기능은 설치된 FC HBA(예: QLogic 및 Emulex)의 작동 매개변수를 관리하고 구성하는 수단을 제공합니다. modprobe 기능은 /etc/modprobe.d 디렉터리 내의 텍스트 파일을 편집(또는 생성)하여 구성됩니다

3.1.1 구성

QLogic 및 Emulex HBA 는 /etc/modprobe.d 디렉터리에있는 개별 텍스트 파일 내에 작성된 구성 구문으로 관리됩니다. 이 파일들이 아직 없으면 생성해야 합니다. QLogic 및 Emulex 하드웨어의 구성 구문은 각각 3.6 "[다중 경로 환경](#)" 및 3.7 "[단일 경로 환경](#)" 섹션에서 자세히 설명합니다. 새로운 RHEL 7x 설치에 이 파일이 없으면 텍스트 편집기를 사용하여 수동으로 만들 수 있습니다.

- Qlogic 의 경우:

```
/etc/modprobe.d/qla2xxx.conf
```

- Emulex 의 경우:

```
/etc/modprobe.d/lpfc.conf
```

3.1.2 modprobe 및 mkinitrd 다시 로드

구성을 변경한 후에 modprobe 기능을 다시 로드하여 새 구성 또는 업데이트된 구성을 적용합니다

로컬 부트 시스템의 경우 모든 SAN 볼륨을 마운트 해제한 다음 모듈을 다시 로드하는 것이 좋습니다 아래에 표시된 대로 모듈을 다시 로드하기 전에 메모리에서 모듈을 언로드해야 합니다(QLogic 의 경우).

```
# modprobe -r qla2xxx
# modprobe qla2xxx
```

Emulex 하드웨어로 작업하는 경우 **qla2xxx** 를 **lpfc** 로 바꿉니다. 이후에 SAN 볼륨을 다시 마운트할 수 있습니다.

SAN 시스템에서 부팅할 때 initramfs-*.img 파일도 재구성해야 부팅 중 새로운 구성이 통합됩니다 아래의 방법은 initramfs-*.img 파일을 다시 작성하는 방법을 보여줍니다 이 프로세스는 기존 위치에서 동일한 파일을 덮어씁니다. 이 절차를 적용하기 전에 기존 initramfs-*.img 파일을 복사하고 백업하는 것이 좋습니다. 다음과 같이 두 명령을 사용할 수 있습니다.

```
# cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).img.bak
# dracut -f
```



3.2 iSCSI

RHEL 7x 는 오픈 iSCSI 스택(RFC3720 기반)의 업데이트된 소프트웨어 구현을 도입했습니다. 이 새로운 iSCSI 스택은 사용자 공간 대신 커널 메모리 공간에 있습니다.

iSCSI 는 기업이 기존 인프라를 활용하면서 엔터프라이즈 저장소 영역까지 확장할 수 있도록 해주는 성숙한 기술입니다. 이 섹션에서는 오픈 iSCSI 스택의 구성, 사용 및 구현에 대해서만 설명합니다. 다른 고급 iSCSI 소프트웨어 구현(사용자 지정 iSCSI HBA 드라이버 및 오프로드 엔진 포함)은 이 백서의 범위를 벗어납니다. 이에 따라 관련 공급업체의 설명서를 참조하십시오.

3.2.1 구성

구성된 Linux 호스트에는 Storage Center 의 iSCSI 포트와 통신할 수 있는 네트워크 포트가 필요합니다. 이 목적으로 전용 포트/VLAN 을 사용하는 것이 좋습니다.

iSCSI 를 구성할 때 더 중요한 고려 사항 중 하나가 네트워크 경로입니다. 시간을 내서 iSCSI 트래픽에 필요한 민감도, 기밀성, 보안 및 대기 시간을 결정하십시오. 이러한 요구에 따라 iSCSI 아키텍처의 네트워크 토폴로지(예: 전용 물리적 포트, VLAN 사용, 다중 경로 및 중복성)를 유도하고 정의합니다.

이상적인 시나리오에서는 iSCSI 트래픽이 전용 포트, 스위치 및 인프라를 사용하여 일상적인 네트워크 트래픽과 분리 및 격리됩니다. 물리적 토폴로지가 제약 사항인 경우 VLAN 서브넷을 사용하여 iSCSI 트래픽을 분리 및 격리하는 것이 좋습니다. 또한 다중 경로 구성에서 항상 iSCSI 를 사용하여 경로 중복성을 생성하는 것이 좋습니다.

VLAN 서브넷을 사용할 수 없는 경우 다음 두 가지 옵션을 선택해야 합니다.

- 정적 라우트를 정의하여 네트워크 계층에서 트래픽 라우팅
- 구성을 통해 iSCSI 레벨에서 트래픽 라우팅

다음 데모에서는 Linux 호스트의 Storage Center 검색 IP 에서 대상을 요청하고 이러한 대상에 로그인한 후 Storage Center 에서 iSCSI 기반 서버 개체를 설정하는 과정을 설명합니다. 새로 발견된 볼륨을 스캔하는 것과 같은 추가 방법은 2.1 "[새 볼륨 스캔](#)" 섹션에서 설명합니다.

이 시나리오에서는 데모 목적으로 아래 ifconfig 및 netstat 출력에 표시된대로 em2 인터페이스만 iSCSI VLAN 에 구성됩니다. 프로덕션 환경에서는 경로, 중복성 및 I/O 큐 관리를 위해 dm-multipath 드라이버를 사용하여 Linux 호스트 당 둘 이상의 iSCSI 인터페이스를 구성하는 것이 좋습니다.

1. 네트워크 인터페이스를 구성합니다.

```
# ifconfig -a
[snip]
em2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.83.21 netmask 255.255.0.0 broadcast 10.10.255.255
    ether 14:fe:b5:c9:95:7f txqueuelen 1000 (Ethernet)
    RX packets 12670048 bytes 1031419587 (983.6 MiB)
    RX errors 0 dropped 34235 overruns 0 frame 0
    TX packets 59 bytes 9454 (9.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
[snip]

# netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask          Flags      MSS Window  irtt Iface
0.0.0.0          172.16.16.1     0.0.0.0          UG         0 0        0 em1
10.10.0.0        0.0.0.0         255.255.0.0      U          0 0        0 em2
172.16.16.0      0.0.0.0         255.255.240.0    U          0 0        0 em1
224.0.0.0        -               255.255.255.0    !          - -        - -
255.255.255.255 -               255.255.255.255 !H         - -        - -
```

2. em2 인터페이스가 구성되어 있고 Storage Center 검색 IP 주소가 10.10.97.2 및 10.10.130.2 이므로(Storage Center 관리 GUI 에서 확인한대로)이 IP 주소를 ping 할 수 있는지 확인하십시오.

```
# ping 10.10.97.2
PING 10.10.97.2 (10.10.97.2) 56(84) bytes of data.
64 bytes from 10.10.97.2: icmp_seq=1 ttl=64 time=0.289 ms
64 bytes from 10.10.97.2: icmp_seq=2 ttl=64 time=0.353 ms
[snip]

# ping 10.10.130.2
PING 10.10.130.2 (10.10.130.2) 56(84) bytes of data.
64 bytes from 10.10.130.2: icmp_seq=1 ttl=64 time=0.394 ms
64 bytes from 10.10.130.2: icmp_seq=2 ttl=64 time=0.318 ms
[snip]
```

3. 다음 명령을 실행하여 Storage Center 대상 IQN 을 요청하고 Linux 호스트가 이 대상에 로그인하도록 요청하십시오.

```
# iscsiadm -m discovery -t sendtargets -p 10.10.97.2:3260
10.10.97.1:3260,0 iqn.2002-03.com.compellent:5000d3100002bb2f
10.10.97.1:3260,0 iqn.2002-03.com.compellent:5000d3100002bb30
10.10.97.1:3260,0 iqn.2002-03.com.compellent:5000d3100002bb32
10.10.97.1:3260,0 iqn.2002-03.com.compellent:5000d3100002bb3b
```

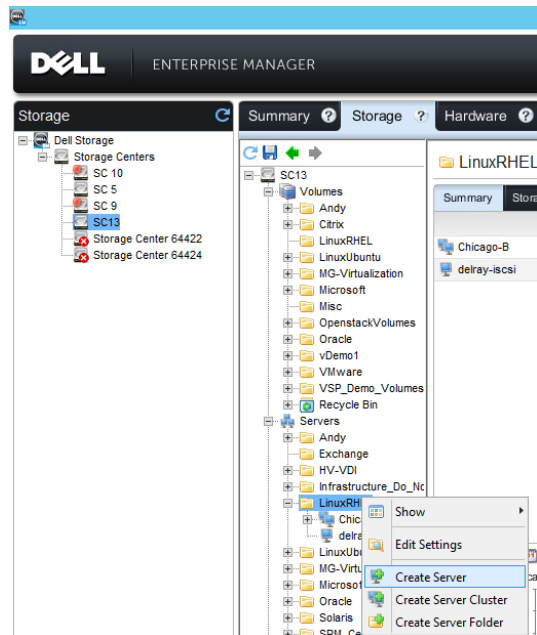


```
# iscsiadm -m discovery -t sendtargets -p 10.10.130.2:3260
10.10.130.1:3260,0 iqn.2002-03.com.compellent:5000d3100002bb31
10.10.130.1:3260,0 iqn.2002-03.com.compellent:5000d3100002bb39
10.10.130.1:3260,0 iqn.2002-03.com.compellent:5000d3100002bb3a
10.10.130.1:3260,0 iqn.2002-03.com.compellent:5000d3100002bb3c

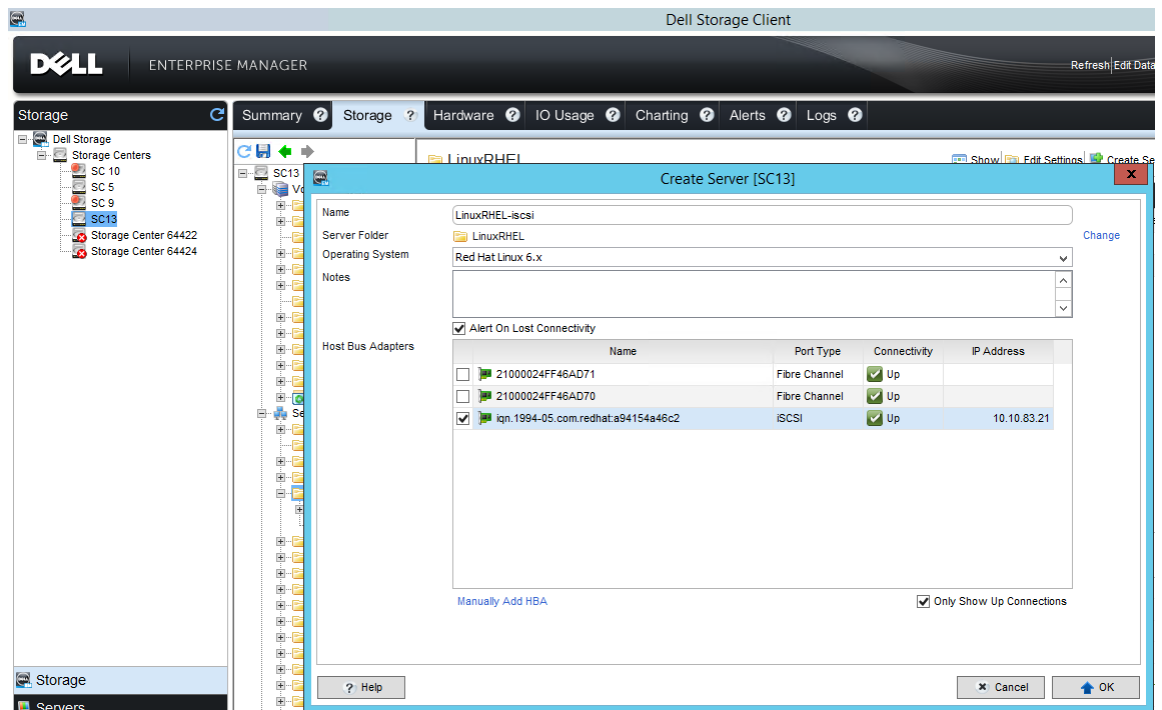
# iscsiadm -m node --login
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d3100002bb2f, portal: 10.10.97.1,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d3100002bb30, portal: 10.10.97.1,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d3100002bb32, portal: 10.10.97.1,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d3100002bb3b, portal: 10.10.97.1,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d3100002bb31, portal: 10.10.130.1,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d3100002bb39, portal: 10.10.130.1,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d3100002bb3a, portal: 10.10.130.1,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d3100002bb3c, portal: 10.10.130.1,3260] (multiple)
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d3100002bb2f,
portal: 10.10.97.1,3260] successful.
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d3100002bb30,
portal: 10.10.97.1,3260] successful.
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d3100002bb32,
portal: 10.10.97.1,3260] successful.
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d3100002bb3b,
portal: 10.10.97.1,3260] successful.
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d3100002bb31,
portal: 10.10.130.1,3260] successful.
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d3100002bb39,
portal: 10.10.130.1,3260] successful.
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d3100002bb3a,
portal: 10.10.130.1,3260] successful.
Login to [iface: default, target: iqn.2002-03.com.compellent:5000d3100002bb3c,
portal: 10.10.130.1,3260] successful.
```



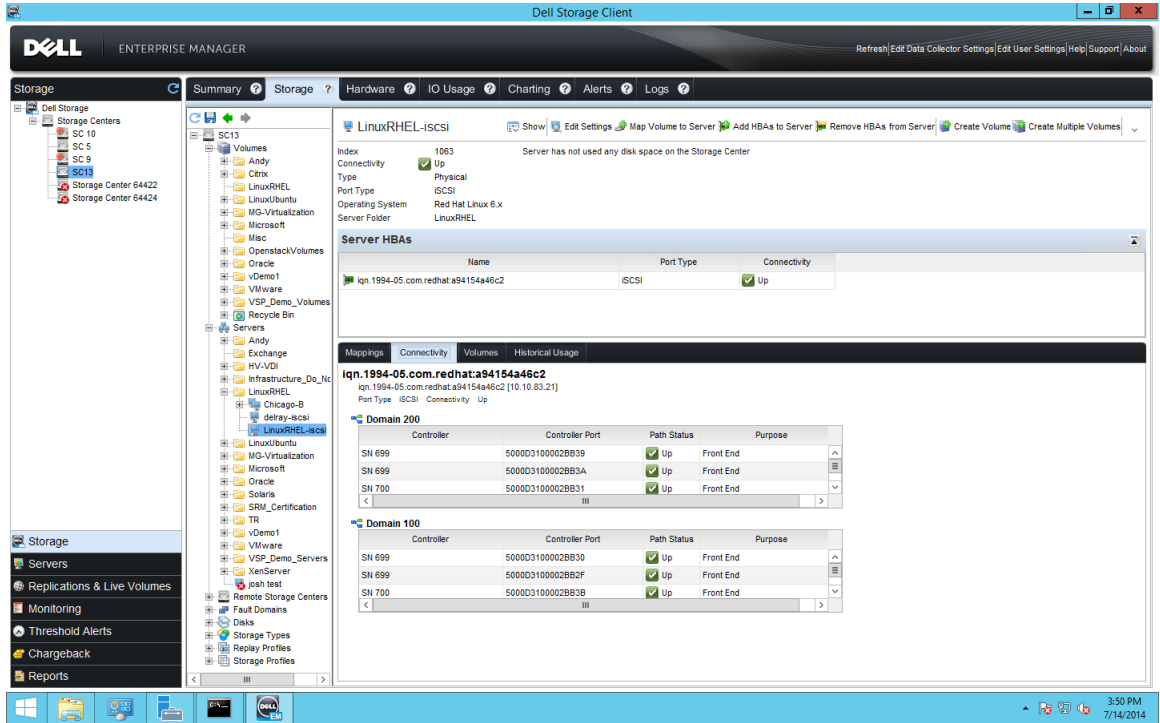
4. Storage Center 에서 새 iSCSI 서버 개체를 설정합니다.
 - a. 해당 서버 트리 폴더를 마우스 오른쪽 단추로 클릭한 다음 **Create Server(서버 생성)**를 클릭합니다.



5. 서버 개체의 이름을 지정하고 운영 체제를 **Red Hat Linux 6.x** 로 설정하고 Linux 호스트 IP 주소(이 경우 10.10.83.21)에 해당하는 IQN 확인란을 선택한 다음 **OK(확인)**를 클릭합니다.



새로운 **LinuxRHEL-iscsi** 서버 개체는 LinuxRHEL 폴더에 표시되며 볼륨을 매핑할 준비가 되었습니다.



3.2.2 iSCSI 시간 초과

dm-multipath 에 의해 경로가 관리되는 다중 경로 iSCSI 환경에서 5 초마다 (각 대상에 NOP-Out 요청을 보내서) SAN 문제를 모니터링하도록 iSCSI 를 구성하십시오. 또한 이 요청을 5 초마다 시간 초과합니다. 이 기능은 I/O 요청을 대체 및 활성 iSCSI 경로에 자동으로 다시 제출함으로써 단일 경로가 복구되기를 기다리는 데 걸리는 대기 시간을 관리하고 최소화합니다.

대부분의 SAN 기반 볼륨(Storage Center 볼륨 포함)의 컴파일된 기본값에는 모든 경로를 사용할 수 없으면 모든 I/O 요청을 자동으로 대기 및 재시도하도록 dm-multipath 에 지시하는 `features='1 queue_if_no_path'` 옵션이 있습니다. 이는 아래 샘플 출력에서 볼 수 있으며 기본 상태로 두어야 합니다.

```
# multipath -ll
[snip]
vol_001 (36000d3100000650000000000000017f2) dm-3 COMPELNT,Compellent Vol
size=50G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
    |- 1:0:3:1 sdf 8:80 active ready running
    |- 1:0:5:1 sdk 8:160 active ready running
    |- 4:0:4:1 sdg 8:96 active ready running
    `-- 4:0:5:1 sdj 8:144 active ready running
[snip]
```



iSCSI 구성 매개변수 NOP-Out 및 연관된 시간 초과 값은 /etc/iscsi/iscsid.conf 파일의 두 항목에 의해 관리됩니다. 두 항목의 기본값은 RHEL 7x 의 기본값 5 입니다.

```
node.conn[0].timeo.noop_out_interval = X
node.conn[0].timeo.noop_out_timeout = X
```

다중 경로 iSCSI 사용의 경우 /etc/iscsi/iscsd.conf 파일의 추가 매개변수는 기본값인 120 에서 5 로 구성해야 합니다. 이렇게 하면 iSCSI 레이어가 이 경로에 대한 명령을 실패하기 전에 최대 5 초 동안 기다려야 하므로 대기 시간 및 I/O 대기를 최소화할 수 있습니다.

```
node.conn[0].timeo.replacement_timeout = 5
```

3.2.3 iSCSI 및 /etc/fstab

iSCSI 는 운영 네트워크에 따라 다르며 /etc/fstab 에 추가된 볼륨은 네트워크 종속으로 지정해야 합니다. 즉, 네트워크 계층 서비스가 시작되고 네트워크가 작동할 때까지 iSCSI 볼륨을 마운트하지 마십시오. 아래 예제는 /etc/fstab 파일에서 **_netdev** 마운트 옵션을 사용하여 iSCSI 마운트에 네트워크 종속성을 만드는 방법을 보여줍니다.

```
# cat /etc/fstab
[snip]
/dev/mapper/VolGroup00-home /home          xfs     defaults          1 2
/dev/mapper/VolGroup00-swap swap      swap     defaults          0 0
LABEL=iSCSI_Vol          /mnt/iSCSI_Vol  ext4    defaults, _netdev 0 0
[snip]
```



3.3 직렬 연결된 SCSI

Dell SCv2000 시리즈 제품 라인은 SAS(Serial Attached SCSI) 프론트 엔드 연결을 제공하며 SCOS 6.6.x 출시와 동시에 사용됩니다. Dell SCv2000 시리즈는 대상 호스트에서 Dell 12Gbps SAS HBA 를 사용할 수 있도록 지원합니다.

Linux 호스트에 대한 SAS 연결에는 /etc/multipath.conf 파일의 특정 구성 스키마가 필요합니다.

3.3.1 SAS 드라이버

SAS 드라이버는 특정 Linux 커널(RHEL 6.5 이상 및 RHEL 7x)에 사전 로드됩니다. 다음 명령을 사용하여 SAS 드라이버의 존재 여부를 확인할 수 있습니다. 가장 좋은 방법은 Dell 저장소 호환성 매트릭스를 사용하여 드라이버 버전의 유효성을 검사하고 표시된 지원되는 최신 드라이버를 사용하는 것입니다.

```
# lsmod | grep sas
mpt3sas                188001  4
scsi_transport_sas     35588  1 mpt3sas
raid_class             4388  1 mpt3sas
megaraid_sas           96205  2

# modinfo mpt3sas | grep description
description:    LSI MPT Fusion SAS 3.0 Device Driver
```

3.3.2 SAS /etc/multipath.conf

다음 장치 스키마를 /etc/multipath.conf 파일에 추가하고 SAS 가 연결된 Linux 호스트에만 사용됩니다. 이 스키마는 vendor="COMPELNT" 및 product="Compellent Vol"로 식별되는 모든 장치에 대한 구성 매개변수를 정의합니다. 이 스키마는 일반적으로 defaults 스키마 뒤에 추가되고 blacklist_exceptions 스키마 앞에 추가됩니다.

```
devices {

    device {
        vendor COMPELNT
        product "Compellent Vol"
        path_checker tur
        prio alua
        path_selector "service-time 0"
        path_grouping_policy group_by_prio
        no_path_retry 24
        hardware_handler "1 alua"
        failback immediate
        rr_weight priorities
    }

}
```



3.3.3 FC/iSCSI 및 SAS

동일한 Linux 호스트에서 병합된 FC/SCSI 및 SAS 연결 환경을 사용하는 것은 검증되거나 지원되지 않습니다. FC/iSCSI 연결을 사용하여 다른 Linux 호스트와 별개인 SAS 연결 Linux 호스트를 유지 관리합니다.

SAS 고유의 multipath.conf 구성 스키마는 FC/iSCSI 기반 /etc/multipath.conf 파일과 병합할 수 있습니다. 이 병합된 구성은 Dell 에서 지원하지 않습니다.

3.3.4 Linux 에서 SAS 장치 식별

이 샘플 스크립트는 Linux /sys 파일 시스템을 구문 분석하고 host_sas_address 로 식별된 파일의 내용을 표시합니다. 이 파일의 내용은 설치된 SAS HBA 의 장치 이름을 나타냅니다. 이 스크립트는 명시된 보증이나 어떠한 종류의 지원 없이 있는 그대로 제공됩니다.

아래의 예에서 두 개의 SAS HBA 카드는 장치 이름이 표시된대로 식별됩니다. 이 스크립트는 RHEL 6x 및 RHEL 7x Linux 호스트에서 작동합니다.

```
# for i in `find /sys/devices -name host_sas_address`; do echo "=== $i"; cat $i;
echo; done
===
/sys/devices/pci0000:40/0000:40:01.0/0000:41:00.0/host1/scsi_host/host1/host_sas
_address
0x544a842007902800

===
/sys/devices/pci0000:40/0000:40:03.0/0000:42:00.0/host2/scsi_host/host2/host_sas
_address
0x544a84200792ce00
```

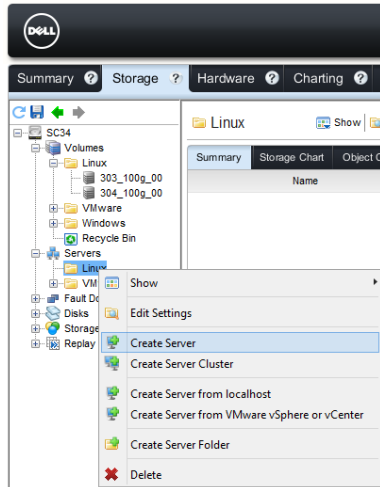


3.3.5 Dell SCv2000 의 SAS 장치 식별

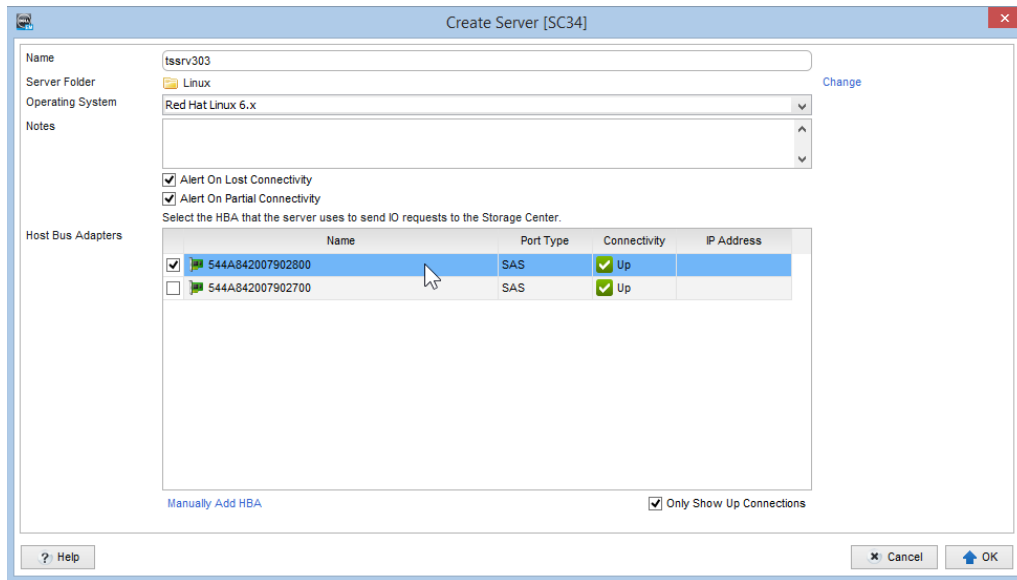
SAS 장치는 여러 SAS WWPN 대신 단일 장치 이름을 통해 Dell SCv2000 에 노출됩니다.

다음 스크린샷 순서는 Dell SC2000 저장소 어레이에 새 서버 개체를 생성하는 절차를 요약한 것입니다

1. **Servers(서버)** 폴더 트리를 마우스 오른쪽 단추로 클릭하고 **Create Server(서버 생성)**를 선택합니다.



2. 새 서버 개체 이름을 지정하고 적절한 운영 체제(Red Hat Linux 6.x 또는 7.x)를 선택하고 Linux 호스트에 해당하는 식별된 SAS 장치 이름을 선택한 다음, **OK(확인)**를 클릭합니다.



3. 새 서버 개체(tssrv303)는 서버 폴더 트리에 작성되며, Linux 호스트가 Storage Center 어레이의 맨 아래 및 맨 위 컨트롤러에 대한 연결을 설정했음을 표시합니다.

The screenshot shows the Dell Storage Center console interface. On the left, a tree view shows the hierarchy: SC34 > Volumes > Linux > tssrv303. The main panel displays the configuration for server tssrv303. It shows the server has not used any disk space on the Storage Center. The 'Server HBAs' section shows a single HBA with WWN 544A842007902800, Port Type SAS, and Connectivity Up. The 'Mappings' section shows the HBA is mapped to SAS Domain 1. The 'Connectivity' section shows the HBA is connected to the Bottom Controller and Top Controller, both with Path Status Up.

Name	Port Type	Connectivity
544A842007902800	SAS	Up

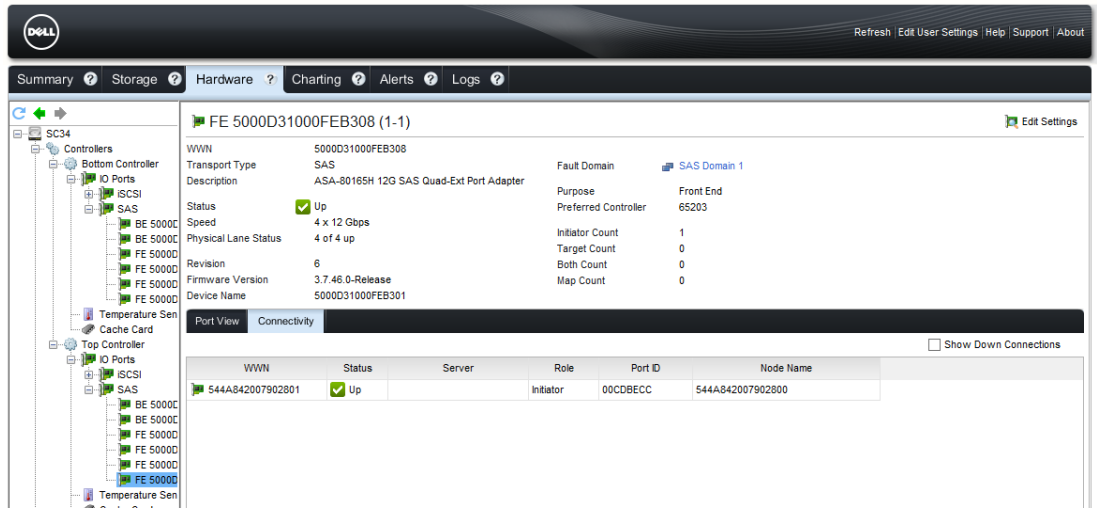
Controller	Controller Port	Path Status	Slot	Slot Port	Fault Domains
Bottom Controller	FE 5000D31000FEB315 (1-1)	Up	1	1	
Top Controller	FE 5000D31000FEB308 (1-1)	Up	1	1	

4. 하드웨어 창에서 *FEB315 로 끝나는 SAS 컨트롤러를 선택하여 연결 창에서 *2800 으로 끝나는 실제 SAS WWPN 을 표시하십시오.

The screenshot shows the Dell Storage Center console interface. On the left, a tree view shows the hierarchy: SC34 > Controllers > IO Ports > SAS > FE 5000D31000FEB315 (1-1). The main panel displays the configuration for the SAS controller FE 5000D31000FEB315 (1-1). It shows the controller is Up, with Speed 4 x 12 Gbps, Physical Lane Status 4 of 4 up, Revision 6, Firmware Version 3.7.46.0-Release, and Device Name 5000D31000FEB302. The 'Port View' section shows the controller is connected to the Bottom Controller and Top Controller, both with Path Status Up.

WWN	Status	Server	Role	Port ID	Node Name
544A842007902800	Up	tssrv303	Initiator	001699BB	544A842007902800

5. 하드웨어 창에서 *FEB308 로 끝나는 SAS 컨트롤러를 선택하여 연결 창에서 *2801 로 끝나는 실제 SAS WWPN 을 표시하십시오.



3.3.6 구성된 다중 경로

제대로 구성된 SAS 볼륨은 다음 "multipath -ll" 출력을 반환합니다. 이 Storage Center 볼륨은 각 경로에서 I/O 가 가능한 다중 경로 ALUA 가능 볼륨으로 검색됩니다. prio=50(활성/최적화 경로)으로 표시된 경로는 모든 활성 I/O 요청에 사용됩니다. prio=1(대기 경로)로 표시되는 경로는 가용성이 높은 중복 경로이며 활성/최적화 경로를 사용할 수 없는 경우에 사용됩니다.

```
# multipath -ll
Compelnt_0016 (36000d31000feb30000000000000000016) dm-3 COMPELNT,Compellent Vol
size=100G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
|  `-- 1:0:0:1 sdb 8:16 active ready running
`+- policy='service-time 0' prio=1 status=enabled
   `-- 1:0:1:1 sdc 8:32 active ready running
Compelnt_001a (36000d31000feb3000000000000000001a) dm-4 COMPELNT,Compellent Vol
size=100G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|+- policy='service-time 0' prio=50 status=active
|  `-- 1:0:1:2 sdd 8:48 active ready running
`+- policy='service-time 0' prio=1 status=enabled
   `-- 1:0:2:2 sde 8:64 active ready running
```

3.3.7 SAS 큐 크기

현재 지원되는 SAS HBA 는 기본적으로 볼륨 당 큐 크기가 254 입니다. 이 값은 특별히 응용 프로그램 계층 구성 요구 사항에서 지정하지 않는 한 출고 시 기본값으로 두어야 합니다.

```
# lsscsi -L | egrep 'COMPELNT|depth'
[1:0:0:1]    disk      COMPELNT Compellent Vol    0606   /dev/sdb
             queue_depth=254
[1:0:1:1]    disk      COMPELNT Compellent Vol    0606   /dev/sdc
             queue_depth=254
```

3.3.8 SAS 에서 부팅

Linux 에서 Dell 12Gbps SAS HBA 와 함께 사용할 때 SAS 에서 부팅 기능이 유효하지 않거나 지원되지 않습니다.

3.4 큐 크기 관리

파이버 채널 HBA 의 큐 크기를 구성하는 위치는 두 곳입니다. 처음 구성할 수 있는 위치는 HBA 의 BIOS 구성 메뉴입니다. 이 구성 메뉴는 일반적으로 부팅하는 동안 또는 HBA 공급업체가 제공하는 도구를 사용하여 액세스합니다. 큐 크기는 운영 체제 내의 Linux modprobe 기능을 통해 관리되는 HBA 구성 파일을 사용하여 구성할 수도 있습니다. 이 두 위치에서 큐 크기 값이 다른 경우 두 값 중 작은 값이 우선합니다. BIOS 를 높은 값(8Gb QLogic HBA 와 함께 사용할 경우 이 값을 실행 스토틀이라고 하며 기본값은 65535)으로 구성한 후 해당 운영 체제 구성 파일로 HBA 큐 크기를 관리하는 것이 좋습니다. 권장되는 시작점은 큐 크기 값 128 입니다. 개별 환경 요구를 충족시킬 수 있도록 값을 평가, 테스트 및 조정하십시오.

- Qlogic 의 경우:

```
options qla2xxx ql2xmaxqdepth=<value>
```

- Emulex 의 경우:

```
options lpfc lpfc_hba_queue_depth=<value> lpfc_lun_queue_depth=<value>
```

이 파일 안의 값은 완전한 구성 세트를 정의하는 데 필요한 다른 추가 키 값 쌍과 스택될 수 있습니다.



3.5 /etc/multipath.conf 및 Storage Center 장치 정의

Storage Center 장치 정의는 커널에 통합됩니다. 아래와 같이 기본 구문 제약 조건 내에서 /etc/multipath.conf 를 사용하십시오. 이 예에서 다른 모든 공급업체 장치는 블랙리스트에 있지만 Storage Center 장치는 고유한 WWID 값을 통해 blacklist_exceptions 절에서 식별됩니다. 이 /etc/multipath.conf 구성은 FC 및 iSCSI 구현에만 적용됩니다. SAS /etc/multipath.conf 구성은 [5.3.2 섹션](#)에서 논의되고 SAS 구현에서 대신 사용되어야 합니다.

```
# cat /etc/multipath.conf
# multipath.conf written by anaconda

defaults {
    user_friendly_names yes
}

blacklist {
    devnode "^ (ram|raw|loop|fd|md|dm-|sr|scd|st) [0-9]*"
    devnode "^hd[a-z]"
    devnode "^dcssblk[0-9]*"
    device {
        vendor "DGC"
        product "LUNZ"
    }
    device {
        vendor "IBM"
        product "S/390.*"
    }
    # don't count normal SATA devices as multipaths
    device {
        vendor "ATA"
    }
    # don't count 3ware devices as multipaths
    device {
        vendor "3ware"
    }
    device {
        vendor "AMCC"
    }
    # nor highpoint devices
    device {
        vendor "HPT"
    }
    wwid "20080519"
    wwid "20080519"
    device {
        vendor iDRAC
    }
}
```



```

        product Virtual_CD
    }
    device {
        vendor PLDS
        product DVD-ROM_DS-8D3SH
    }
    #wwid "*"
}

blacklist_exceptions {
    device {
        vendor "COMPELNT"
        product "Compellent Vol"
    }
}

multipaths {
    multipath {
        alias vol_001
        wwid "36000d31000006500000000000000017f2"
        uid 0
        gid 0
        mode 0600
    }
    multipath {
        alias vol_002
        wwid "36000d31000006500000000000000017f3"
        uid 0
        gid 0
        mode 0600
    }
    multipath {
        alias vol_003
        wwid "36000d31000006500000000000000017f4"
        uid 0
        gid 0
        mode 0600
    }
}

```



3.5.1 다중 경로 별칭

위 예의 /etc/multipath.conf 파일에서 Storage Center WWID 값은 각각 multipaths 절 내의 별칭 값(vol_XXX)으로 식별됩니다. Linux 에서 다중 경로 별칭/이름은 볼륨의 기능을 보다 잘 식별할 뿐만 아니라 재부팅 및 재구성을 통해 이러한 장치 이름의 지속성을 유지합니다. 즉, 다중 경로 별칭/이름은 스크립팅, 마운트 명령, /etc/fstab 구문 등에서 사용하기에 일관되고 안전합니다.

다중 경로 별칭이 /etc/multipath.conf 에 정의되어 있지 않으면 볼륨은 재부팅되는 동안 지속성을 유지하면서 /dev/mapper/mpathX 의 기본 명명 스키마를 기본값으로 사용합니다. 다중 경로 장치 이름을 보다 알기 쉬운 레이블(예: 비즈니스 기능 또는 사용)과 연결할 때 별칭 지정이 권장되고 매우 유용합니다.

/etc/multipath.conf 에 별칭을 정의하거나 구문을 업데이트한 후에는 아래에 표시된대로 변경 사항을 반영하기 위해 multipathd 데몬을 다시 시작해야 합니다.

```
# systemctl restart multipathd.service
```

RHEL 7x에서는 이전 명령 구문을 사용하여 새로운 systemd 서비스 관리 프레임워크에 명령 구문을 전달할 수 있습니다.

```
# service restart multipathd
```

3.6 다중 경로 환경

Storage Center 장애 조치 이벤트 중에 레거시 포트 모드 작동이 발생하면 실패한 컨트롤러의 활성 포트의 WWPN 이 다른 활성 컨트롤러의 동일한 오류 도메인의 예약 포트로 재배치되기 직전 패브릭에서 사라집니다. 가상 포트 모드 경로 장애 조치에서 활성 NPIV 포트의 WWPN 은 동일한 오류 도메인의 다른 활성 NPIV 포트로 재배치됩니다. 컨트롤러 장애 조치의 경우, WWPN 은 다른 활성 컨트롤러의 동일한 오류 도메인의 다른 활성 NPIV 포트로 재배치됩니다. 어떤 장애 조치 시나리오에서든지 Storage Center 는 이러한 변경 사항을 패브릭을 통해 전파하는 데 5~60 초 정도 걸릴 수 있습니다.

다중 경로 연결 시나리오에서 I/O 중단을 완화하려면 포트를 작동 중지/실패로 표시하기 전에 HBA 레벨 코드가 최대 5 초 동안 대기하도록 지시하는 것이 좋습니다. 이를 통해 I/O 요청을 SAN 패브릭을 통해 대체 및 활성 HBA 경로로 신속하게 재배치함으로써 I/O 또는 그 위에 있는 응용 프로그램 스택에 대한 I/O 대기 시간을 최소화합니다. 모든 경로가 중단된 경우, 하나 이상의 경로가 복구될 때까지 dm-multipath 모듈 계층이 I/O 를 큐에 넣기 시작합니다. 이를 통해 Storage Center 가 실패한 포트의 WWPN 을 활성 포트로 재배치하고 SAN 패브릭을 통해 변경 사항을 전파할 수 있는 충분한 시간을 허용합니다. 다중 경로 환경의 구성 구문은 아래 섹션에서 설명합니다.

SAS 연결된 환경에서 두 컨트롤러의 경로는 연결된 호스트(활성/최적화 및 대기)에 제공되지만 활성/최적화 경로만 모든 활성 I/O 에 대해 한 번에 사용됩니다. 활성/최적화 경로를 사용할 수 없게 되면 Storage Center 는 나머지 대기 경로 중 활성/최적화 경로 역할을 수행할 경로를 결정하고 활성 I/O 를 새 활성/최적화 경로로 계속 스트리밍할지 결정합니다.



3.6.1 다중 경로/SCSI 제한 시간 및 큐

대부분의 SAN 기반 볼륨(Storage Center 볼륨 포함)의 컴파일된 기본값에는 모든 경로를 사용할 수 없으면 모든 I/O 요청을 자동으로 대기 및 재시도하도록 dm-multipath 에 지시하는 `features='1 queue_if_no_path'` 옵션이 있습니다. 이는 아래 샘플 출력에서 볼 수 있으며 기본 상태로 두어야 합니다.

```
# multipath -ll
[snip]
vol_001 (36000d31000006500000000000000017f2) dm-3 COMPELNT,Compellent Vol
size=50G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
    |- 1:0:3:1 sdf 8:80 active ready running
    |- 1:0:5:1 sdk 8:160 active ready running
    |- 4:0:4:1 sdg 8:96 active ready running
    `-- 4:0:5:1 sdj 8:144 active ready running
[snip]
```

I/O 가 대기 중이 아니지만 모든 경로를 사용할 수 없으면 대신 I/O 가 빨리 실패하는 비즈니스/응용 프로그램 시나리오에 이 기본값 사용의 예외가 있습니다. 이러한 상황에서는 `/etc/multipath.conf` 에서 전역 장치 수준이나 특정 장치로 I/O 큐를 비활성화하거나 런타임에 특정 장치에 I/O 큐를 적용할 수 있습니다.

전역 장치 수준에서 다음을 `/etc/multipath.conf` 에 삽입한 다음 `multipathd` 서비스를 다시 시작합니다.

```
devices {
    device {
        vendor "COMPELNT"
        product "Compellent Vol"
        features 0
        no_path_retry fail
    }
}
```

이것을 런타임에 특정 장치에 구현하려면 아래에 표시된대로 원하는 다중 경로 장치당 다음 명령을 실행하십시오.

```
# dmsetup message /dev/mapper/vol_001 0 "fail_if_no_path"
```

3.6.2 PortDown 제한 시간

이 설정은 포트와의 연결이 끊어진 후 연결이 끊기기 전에 Linux 시스템이 대기하는 시간을 지정합니다. 그에 따라 `qla2xxx.conf` 또는 `lpfc.conf` 파일에서 다음을 구성하십시오.

- Qlogic 의 경우:
options qla2xxx qlport_down_retry=5
- Emulex 의 경우:
options lpfc lpfc_devloss_tmo=5



3.6.3 매개변수 확인

구성 변경 사항이 적용되었는지 확인하려면 다음 명령을 적용하십시오.

- Qlogic 의 경우:

```
# cat /sys/module/qla2xxx/parameters/qlport_down_retry  
5
```
- Emulex 의 경우:

```
# cat /sys/class/scsi_host/hostX/lpfc_devloss_tmo  
5
```

3.7 단일 경로 환경

Storage Center 장애 조치 이벤트 중에 레거시 포트 모드 작동이 발생하면 실패한 컨트롤러의 활성 포트의 WWPN 이 다른 활성 컨트롤러의 동일한 오류 도메인의 예약 포트로 재배치되기 직전 패브릭에서 사라집니다. 가상 포트 모드 경로 장애 조치에서 활성 NPIV 포트의 WWPN 은 동일한 오류 도메인의 다른 활성 NPIV 포트로 재배치됩니다. 컨트롤러 장애 조치의 경우, WWPN 은 다른 활성 컨트롤러의 동일한 오류 도메인의 활성 NPIV 포트로 재배치됩니다. 어떤 장애 조치 시나리오에서든 Storage Center 는 이러한 변경 사항을 패브릭을 통해 전파하는 데 5~60 초 정도 걸릴 수 있습니다.

단일 경로 연결 시나리오에서 I/O 중단을 완화하려면 포트를 작동 중지/실패로 표시하기 전에 HBA 레벨 코드가 최대 60 초 동안 대기하도록 지시하는 것이 좋습니다. 이를 통해 Storage Center 가 실패한 포트의 WWPN 을 활성 포트로 재배치하고 SAN 패브릭을 통해 변경 사항을 전파할 수 있는 충분한 시간을 허용합니다. 단일 경로 환경에 대한 구성 구문은 아래에 설명되어 있습니다.

3.7.1 PortDown 제한 시간

이 설정은 포트와의 연결이 끊어진 후 연결이 끊기기 전에 Linux 시스템이 대기하는 시간을 지정합니다. 그에 따라 **qla2xxx.conf** 또는 **lpfc.conf** 파일에서 다음을 구성하십시오.

- Qlogic 의 경우:

```
options qla2xxx qlport_down_retry=60
```
- Emulex 의 경우:

```
options lpfc lpfc_devloss_tmo=60
```

3.7.2 매개변수 확인

구성 변경 사항이 적용되었는지 확인하려면 다음 명령을 적용하십시오.

- Qlogic 의 경우:

```
# cat /sys/module/qla2xxx/parameters/qlport_down_retry  
60
```
- Emulex 의 경우:

```
# cat /sys/class/scsi_host/hostX/lpfc_devloss_tmo  
60
```



4 성능 고려 사항

이 섹션에서는 Linux 에서 사용할 수 있는 좀 더 일반적인 성능 조정 옵션과 변수에 관한 일반적인 정보 및 안내를 제공합니다. 이 정보는 모든 내용을 포함하는 것이 아니며 사용된 값을 최종적인 것으로 간주해서는 안 됩니다. 이 섹션에서는 Linux 및 저장소 관리자가 Linux 설치를 미세 조정하고 최적의 성능을 달성하는데 사용할 수 있는 출발점을 제공합니다.

다음 매개변수를 변경하기 전에 현재 환경 워크로드를 잘 알고 있어야 합니다. 환경을 지원하는 일상적인 경험을 토대로 시스템 또는 저장소 관리자의 인식을 평가하는 등 다양한 방법으로 이를 수행할 수 있습니다. DPACK(Dell Performance Analysis Collection Kit)은 DPACK_Support@Dell.com으로 이메일을 보내서 받을 수 있는 무료 툴킷입니다.

Linux 의 성능을 조정할 때 고려해야 할 몇 가지 일반적인 지침은 다음과 같습니다.

- 성능 조정은 과학인 동시에 예술입니다. 성능(특히 I/O)에 영향을 미치는 많은 변수가 있으므로 모든 환경에 대해 특정 값을 권장할 수는 없습니다. 몇 가지 변수로 시작하고 시스템이 조정되면 더 많은 변수 또는 레이어를 추가하십시오. 예를 들어, 단일 경로로 시작하고, 조정된 다음 다중 경로를 추가하십시오.
- 한 번에 하나씩 변경한 다음 성능 모니터링 도구로 성능에 미치는 영향을 테스트, 측정 및 평가한 후에 변경하십시오.
- 가장 좋은 방법은 필요에 따라 변경 사항을 알려진 상태로 되돌릴 수 있도록 원래 설정을 기록해두는 것입니다.
- 가능한 경우 프로덕션 환경이 아닌 환경에서 시스템 조정 원칙(예: 장애 조치)을 적용하고 프로덕션 환경으로 변경 사항을 전파하기 전에 최대한 많은 환경 조건으로 변경 사항의 유효성을 검사하십시오.
- 성능 요구가 현재 구성 설정으로 충족되는 경우 일반적으로 시스템을 불안정하게 만들 수 있는 변경 사항이 적용되지 않도록 설정을 그대로 두는 것이 가장 좋습니다.
- 성능에 가장 효과적인 영향을 줄 수 있도록 조정 가능한 설정을 효과적으로 지정하려면 블록 및 파일 수준 데이터 간의 차이점을 이해해야 합니다. Storage Center 어레이는 블록 기반 저장 장치이지만 iSCSI 전송 메커니즘에 대한 지원을 통해 일반적으로 네트워크 및 파일 수준 조정과 관련된 성능 고려 사항을 도입합니다.
- 변경 사항이 성능에 영향을 미치는지 검증할 때 Dell Storage Manager 의 차트 기능을 활용하여 성능을 추적하십시오. 또한 I/O 성능에 가장 큰 영향(긍정적/부정적)을 미치는 변수를 보다 잘 추적하기 위해 반복 항목 간에 단일 변경을 수행해야 합니다.



4.1 조정 프로파일

RHEL 7x 는 관리자 및 저장소 전문가가 Red Hat Linux 호스트를 조정하는 데 도움이 되는 새로운 도구 세트를 도입했습니다. 이 도구는 두 개의 새로운 명령(**tuned** 및 **tuned-adm**)을 사용하여 찾을 수 있고, 이 명령은 미리 정의된 성능 조정 프로파일 세트를 관리합니다. 다음 출력은 적용할 수 있는 대체 조정 프로파일 목록뿐만 아니라 활성 조정 프로파일(기본값)을 보여줍니다. 현재는 Storage Center 를 구현할 때 기본 조정 프로파일 **throughput-performance** 를 사용하는 것이 좋습니다. 각 조정 프로파일 및 장점에 대한 설명은 본 백서의 범위를 벗어나지만 이 주제에 대한 자세한 설명은

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Performance_Tuning_Guide/sect-Red_Hat_Enterprise_Linux-Performance_Tuning_Guide-Performance_Monitoring_Tools-tuned_and_tuned_adm.html에서 찾을 수 있습니다.

```
# tuned-adm active
Current active profile: throughput-performance
```

```
# tuned-adm list
□□ □□□ □□□:
- balanced
- desktop
- latency-performance
- network-latency
- network-throughput
- powersave
- sap
- throughput-performance
- virtual-guest
- virtual-host
```

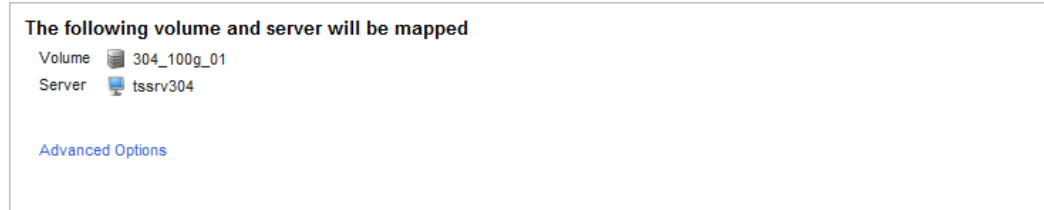
4.2 다중 볼륨

Storage Center 볼륨은 한 번에 하나의 Storage Center 컨트롤러에서만 활성화되어 표시됩니다. 이러한 고려 사항을 염두에 두고 Linux 호스트에서 여러 볼륨을 사용하는 경우 Storage Center 컨트롤러를 통해 Linux 호스트에 볼륨을 번갈아 표시하면 이중 I/O 처리 및 로드를 위해 두 컨트롤러를 가장 효과적으로 활용할 수 있습니다. Linux 호스트에서 이 볼륨을 제거한 LVM 볼륨 그룹에 포함시키면 모든 경로의 병렬 I/O 스트림 수가 Storage Center 어레이에 추가됩니다. 이 방법을 개별 요구, 종종 다양한 비즈니스 환경 및 사용 사례에 맞게 평가하고 적용 및 테스트하십시오.

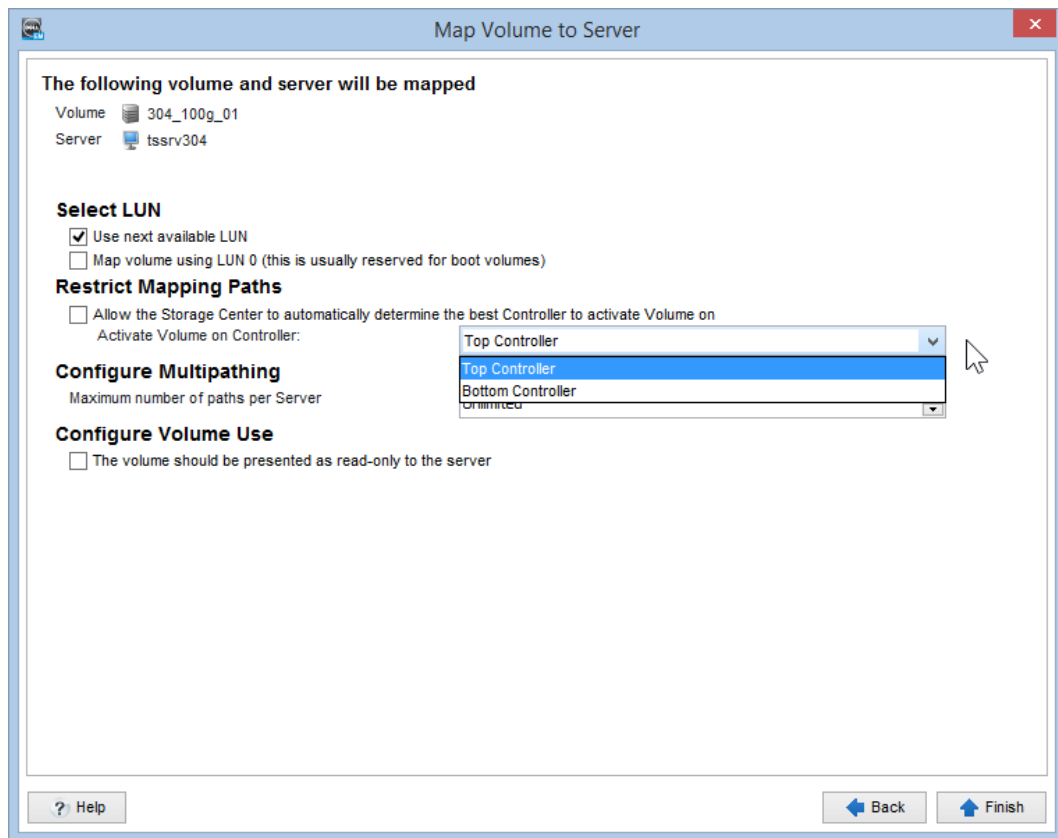
SAS 연결된 환경에서 두 컨트롤러의 경로는 연결된 호스트(활성/최적화 및 대기)에 제공되지만 활성/최적화 경로만 모든 활성 I/O에 대해 한 번에 사용됩니다. 위에서 설명한 방법을 적용하면 각 볼륨에 대해 다른 활성/최적화 경로를 사용하여 여러 컨트롤러에서 얻은 것 보다 더 많은 I/O 스트리밍 및 처리량을 얻을 수 있습니다. 이러한 볼륨을 서버 개체에 매핑할 때 명시적으로 다른 컨트롤러에 볼륨을 고정하면 이를 수행할 수 있습니다. 이 기능은 아래와 같이 매핑 대화 상자의 고급 옵션을 통해 액세스할 수 있습니다.



1. **Advanced Options(고급 옵션)** 링크를 클릭합니다.



2. **Restrict Mapping Paths(매핑 경로 제한)**를 선택 해제하고 서버 개체를 매핑하기 위해 볼륨을 고정할 컨트롤러를 선택하십시오.



4.3 HBA 큐 크기

큐 크기는 보류중인 I/O 요청 수를 나타냅니다. 이 값을 수정하면 특정 워크로드에서 I/O 성능이 향상될 수 있습니다. 일반적으로 큐 크기를 늘리면 처리량이 늘어날 수 있지만 이 값을 높이면 I/O 대기 시간이 길어질 수 있으므로 고려해야 합니다. 대량의 I/O가 작은 읽기 및 쓰기 작업인 환경과 같이 다양한 응용 프로그램에서는 이 값을 늘리는 것이 좋습니다. IOPS 요구 사항은 낮지만 더 높은 처리량이 필요한 환경에서는 최적의 성능 수준이 달성될 때까지 이 큐 크기 설정을 낮추면 됩니다.

이 값은 HBA 펌웨어 또는 HBA용 Linux 커널 모듈에서 변경할 수 있습니다. 이 두 설정의 값이 다른 경우 낮은 값이 우선합니다. 고려해야 할 좋은 전략은 HBA 펌웨어를 허용 가능한 최대 수로 설정한 다음 이 값을 Linux 커널 모듈 내에서 아래로 조정하는 것입니다.

사용되는 특정 공급업체 HBA에 대해 이 값을 수정하는 방법에 대한 자세한 내용은 3" [서버 구성](#)" 섹션을 참조하십시오.

4.4 SCSI UNMAP/TRIM

파일 시스템 사용 사례에서의 SCSI UNMAP/TRIM의 적용과 사용은 기업의 저장소 및 비용 관리에 매우 효과적일 수 있습니다. 그러나 이 기능을 구현하는 방법에 대해서도 고려해야 합니다.

discard 마운트 매개변수를 사용하면 파일 시스템이 Storage Center 어레이에 대해 실시간/즉각적으로 SCSI UNMAP 명령을 수행할 수 있습니다. 높은 I/O 기반 응용 프로그램 환경에서는 불필요하게 높은 수준의 I/O 제어 트래픽이 발생하고 CPU 로드가 상승하여 I/O 대기 시간이 늘어나고 업무에 중요한 I/O가 영향을 받을 수 있습니다.

이 구현의 대안은 fstrim 명령을 사용하는 것입니다. fstrim 명령은 util-linux 패키지의 일부이며 마운트된 파일 시스템에서 사용된 블록을 한 번에 삭제할 수 있습니다. 이 명령은 스크립팅(아래에 표시된 샘플 스크립트)되어 예약된 cron 작업에 삽입된 다음, 비즈니스 핵심 기능에 덜 영향을 줄 수 있는 시간대에 일괄적으로 마운트된 파일 시스템 세트에 적용될 수 있습니다.

```
#!/bin/bash

FSTrim=/usr/bin/fstrim

MntPoints="u01 u02 u03 u04 u05"

for i in ${MntPoints}
do
    echo "INFO: Applying ${FSTrim} to mount point ${i}"
    ${FSTrim} -v /${i}
done
```



4.5 Linux SCSI 장치 큐 변수

여러 Linux SCSI 장치 큐 설정을 조정하여 성능을 향상시킬 수 있습니다. 가장 일반적인 매개변수는 I/O와 관련하여 각 매개변수가 수행하는 작업에 대한 간략한 설명과 함께 섹션에 4.5.14.5.3 나와 있습니다. 이 값은 `/sys/block/dm-X/queue` 디렉터리(다중 경로 장치의 경우) 및 `/sys/block/dm-X/queue` 디렉터리(블록 장치의 경우)에 있으며 성능 수정을 목표로 하는 볼륨의 각 장치에 대해 수정해야 합니다.

4.5.1 커널 I/O 스케줄러

`/sys/block/dm-X/queue/scheduler` 매개변수와 그 내용은 SCSI(sd) 장치 용 Linux 커널에서 사용 중인 I/O 스케줄러를 정의합니다. 일부 응용 프로그램 공급업체(예: Oracle)는 응용 프로그램 플랫폼에서 최적의 성능을 얻기 위해 I/O 스케줄러에 대한 특정 권장 사항을 제공합니다. RHEL 7x 에서 기본적으로 이 변수는 파일 내에서 **[]** 대괄호로 표시된 **deadline** 으로 설정됩니다. **deadline** 스케줄러는

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Performance_Tuning_Guide/chap-Red_Hat_Enterprise_Linux-Performance_Tuning_Guide-Storage_and_File_Systems.html#sect-Red_Hat_Enterprise_Linux-Performance_Tuning_Guide-Considerations-IO_Schedulers 의 Red Hat 사이트에서 설명한 sysfs 조정 가능 하위 매개변수(예: `fifo_batch`, `read_expire` 및 `write_expire`)를 사용하여 추가로 구성할 수 있습니다.

```
# cat /sys/block/dm-X/queue/scheduler
noop [deadline] cfq
```

이 매개변수는 다음 명령을 수행하여 동적으로 변경할 수 있습니다. 그러나 스케줄러를 모든 Storage Center 구현에 대한 **deadline** 으로 남겨 두는 것이 좋습니다. 또한 스케줄러 값은 앞에서 설명한대로 RHEL 7x 조정 프로필 사용에 따라 동적으로 조정됩니다.

```
# echo cfq > /sys/block/dm-X/queue/scheduler
# cat /sys/block/dm-X/queue/scheduler noop deadline [cfq]
```

스케줄러는 변경된 경우 운영 체제의 현재 실행 중인 인스턴스에만 적용되고 적용되는 특정 `/dev/sdX` 장치에만 적용됩니다. 부팅하는 동안 모든 필수 SCSI(sd) 장치에서(장치 단위로) 이 변경 사항을 지속하도록 스크립트를 추가로 사용할 수 있습니다. 또는 아래와 같이 `/boot/grub2/grub.cfg` 부팅 구성 파일 내부의 커널 문자열 끝에 `elevator=key value` 옵션을 추가하여 부팅 시 시스템 전체에 적용할 수 있습니다.

```
linux16 /vmlinuz-0-rescue-c417dfc159fc4450ac2cc28137506041 root=UUID=35bea1c0-
ce32-42a4-8e36-72fd5e77471d ro rd.lvm.lv=VolGroup00/root
vconsole.font=latacyrheb-sun16 rd.lvm.lv=VolGroup00/swap crashkernel=auto
vconsole.keymap=us rhgb quiet elevator=deadline
```



4.5.2 read_ahead_kb

이 매개변수는 커널이 블록 장치에서 순차적으로 읽기를 감지할 때 사용되고 Linux 커널이 읽는 I/O의 킬로바이트 수를 정의합니다. 이 값을 수정하면 대량의 순차 읽기 워크로드에서 성능에 큰 영향을 미칠 수 있습니다. 이 매개변수의 RHEL 7x 기본값은 4096입니다. 이 기본 상태가 양호한 시작점입니다.

4.5.3 nr_requests

nr_requests 값은 Linux 커널이 요청 큐의 깊이를 설정하는 데 사용되며 HBA의 큐 크기 구성 변경과 함께 사용됩니다. RHEL 7x 기본값은 128입니다. 이 값을 늘리면 I/O 하위 시스템을 더 큰 임계값으로 설정하여 요청을 계속 예약합니다. 이렇게 하면 I/O 하위 시스템이 한 방향으로 더 오랫동안 이동하게 되어 디스크 I/O를 보다 효율적으로 처리할 수 있습니다. 이 값을 1024로 높이고 평가한 다음, 성능 결과에 따라 조정하십시오.

4.6 iSCSI

iSCSI 조정은 이더넷 네트워크와 블록 레벨 조정 모두에 대한 작업입니다. 일반적인 이더넷 커널 조정 가능 매개변수를 평가하여 iSCSI에서 최적의 성능 향상을 제공하는 설정을 결정하십시오. 정보 프레임을 사용하면 1Gb/10Gb 이더넷과 함께 사용할 때 iSCSI 성능이 향상될 수 있습니다. Fibre Channel과 마찬가지로 전체 성능에 대한 영향을 완전히 이해하려면 iSCSI 변경을 개별적, 점진적으로 여러 워크로드 유형에 대해 평가해야 합니다. 즉, 네트워크(이더넷) 조정 외에도 블록 수준의 하위 시스템 조정 고려 사항 때문에 iSCSI 조정에 더 많은 시간이 소요되는 경우가 많습니다. 시스템을 효과적으로 조정하려면 다양한 Linux 하위 시스템 계층에 대한 확실한 이해가 필요합니다.

성능을 위해 조정할 수 있는 커널 매개변수는 `/proc/sys/net/core` 및 `/proc/sys/net/ipv4` 커널 매개변수에서 찾을 수 있습니다. 최적의 값이 결정되면 `/etc/sysctl.conf` 파일에 영구적으로 값을 설정하십시오. 다른 대부분의 최신 운영 체제 플랫폼과 마찬가지로 Linux는 TCP 버퍼를 효율적으로 자동 조정할 수 있습니다. 그러나 기본적으로 일부 설정은 보수적으로 낮습니다. 다음 커널 매개변수를 사용하면 네트워크 성능이 향상되고 이후에 iSCSI 성능이 향상될 수 있습니다.

- TCP □□ □□ □□:
 - net.core.rmem_max
 - net.core.wmem_max
- Linux □□ □□ □□ □□:
 - net.ipv4.tcp_rmem
 - net.ipv4.tcp_wmem
- net.ipv4.tcp_window_scaling
- net.ipv4.tcp_timestamps
- net.ipv4.tcp_sack



5 유용한 도구

다음 고유한 Linux 도구를 사용하여 볼륨을 식별하고 해당 Storage Center 장치와 상호 연관시킬 수 있습니다.

5.1 lsscsi 명령

lsscsi 명령은 /proc 및 /sys 의사 파일 시스템의 정보를 사용자가 읽을 수 있는 출력으로 구문 분석하는 도구입니다. lsscsi 도구는 다음 명령을 사용하여 설치됩니다.

```
# yum -y install lsscsi
```

grep 명령을 사용하여 Storage Center 장치만 표시하는 lsscsi 출력이 아래에 나와 있습니다. 첫 번째 열에는 각 볼륨에 대한 [host:channel:target:lun] 지정이 표시됩니다. 호스트 번호는 볼륨이 매핑된 로컬 HBA hostX 장치 파일에 해당합니다. 채널 번호는 SCSI 버스 주소이며 항상 0입니다. 대상 번호는 Storage Center 프런트 엔드 포트(대상)와 관련됩니다. LUN 번호는 매핑된 Storage Center 컨트롤러에서 볼륨의 LUN ID를 나타냅니다.

```
# lsscsi | grep COMPELNT | sort -k7
[snip]
[1:0:0:1]    disk      COMPELNT  Compellent Vol  0605  /dev/sdb
[1:0:2:1]    disk      COMPELNT  Compellent Vol  0605  /dev/sdc
[1:0:3:1]    disk      COMPELNT  Compellent Vol  0605  /dev/sdd
[1:0:3:2]    disk      COMPELNT  Compellent Vol  0605  /dev/sde
[1:0:5:1]    disk      COMPELNT  Compellent Vol  0605  /dev/sdf
[1:0:5:2]    disk      COMPELNT  Compellent Vol  0605  /dev/sdg
[4:0:0:1]    disk      COMPELNT  Compellent Vol  0605  /dev/sdh
[4:0:2:1]    disk      COMPELNT  Compellent Vol  0605  /dev/sdi
[4:0:4:1]    disk      COMPELNT  Compellent Vol  0605  /dev/sdj
[4:0:4:2]    disk      COMPELNT  Compellent Vol  0605  /dev/sdk
[4:0:5:1]    disk      COMPELNT  Compellent Vol  0605  /dev/sdl
[4:0:5:2]    disk      COMPELNT  Compellent Vol  0605  /dev/sdm
[snip]
```



5.2 scsi_id 명령

scsi_id 명령(/usr/lib/udev 폴더에 있음)을 사용하여 볼륨의 WWID 를 보고할 수 있습니다. 그런 다음 이 WWID 를 사용하여 볼륨을 해당 Storage Center 장치에 연관시킬 수 있습니다.

다음 스크립트는 명시된 보증이나 어떤 종류의 지원 없이 있는 그대로 제공됩니다. 이 스크립트는 scsi_id 명령을 적용하고 Linux 장치 이름 (/dev/sdX)을 각각의 WWID 값과 연관시킵니다.

```
#!/bin/bash

SCSIID=/usr/lib/udev/scsi_id

#OSMajor=`cat ${ReleaseFile} | awk '{print $7}' | cut -d. -f1`
OSMajor=`uname -r | awk -F. '{print $4}'`

echo "INFO: OS Major Rev. ${OSMajor} detected!"

if [ "${OSMajor}" = "el7" -o "${OSMajor}" = "el6" ]; then
{
    for i in `cat /proc/partitions | awk '{print $4}' | grep sd`
    do
        echo "Device: $i WWID: `${SCSIID} --page=0x83 --whitelisted --
device=/dev/$i`"
    done
} | sort -k4 $1
elif [ "${OSMajor}" = "el5" ]; then
{
    for i in `cat /proc/partitions | awk '{print $4}' | grep sd`
    do
        echo "Device: $i WWID: `${SCSIID} -g -u -s /block/$i`"
    done
} | sort -k4 $1
else
    echo "WARN: OSMajor parameter of unknown value, Exiting"
    exit 1
fi
```

이 스크립트의 샘플 출력은 다음과 같습니다.

```
# ./get_WWID.sh
INFO: OS Major Rev. el7 detected!
[snip]
Device: sdd WWID: 36000d31000006500000000000000017f2
Device: sdf WWID: 36000d31000006500000000000000017f2
Device: sdj WWID: 36000d31000006500000000000000017f2
Device: sdl WWID: 36000d31000006500000000000000017f2
[snip]
```



이 WWID 값은 차례로 Storage Center 장치의 일련 번호 값과 연관됩니다.

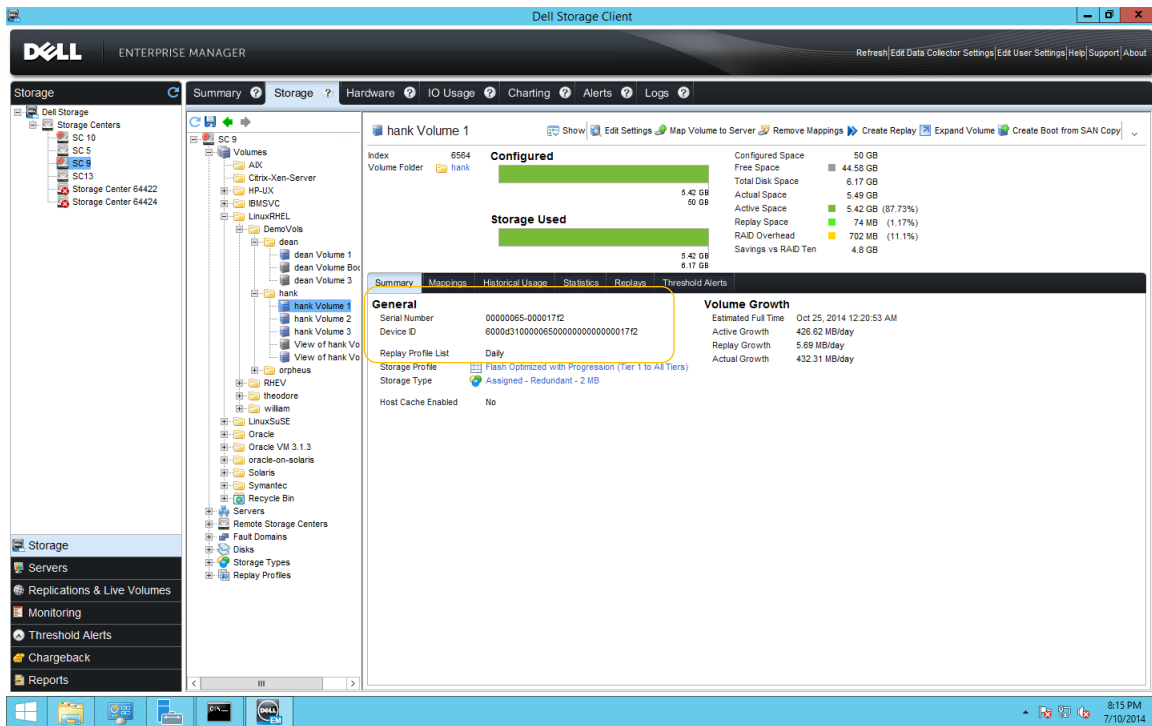


Figure 1 □□□□□□

5.3 dmsetup 명령

dmsetup 명령은 device-mapper 드라이버가 관리하는 논리 볼륨의 메타 데이터를 보고, 관리 및 변경하는 데 사용할 수 있습니다. 아래 예제에서 dmsetup 명령은 Storage Center 볼륨의 값을 표시하는 데 사용됩니다. 이 볼륨의 이름 및 UUID 값(UUID 값은 "mpath-" 접두사와 볼륨 일련 번호 값의 연결입니다).

```
# dmsetup info /dev/mapper/vol_001
Name:          vol_001
State:         ACTIVE
Read Ahead:    256
Tables present: LIVE
Open count:    0
Event number:  0
Major, minor:  253, 4
Number of targets: 1
UUID: mpath-36000d31000006500000000000000017f2
```

5.4 dmesg 명령어

dmesg 명령은 최근에 발견된 볼륨에 할당된 장치 이름을 찾는 데 유용합니다. 아래 출력은 새로운 Storage Center 볼륨을 검색하여 /dev/sdh 장치 파일에 할당하는 방법을 보여줍니다.

```
# dmesg
[snip]
[1203830.267568] scsi 4:0:0:1: Direct-Access      COMPELNT Compellent Vol   0605
PQ: 0 ANSI: 5
[1203830.267996] sd 4:0:0:1: Attached scsi generic sg8 type 0
[1203830.268089] sd 4:0:0:1: [sdh] 20971520 512-byte logical blocks: (10.7
GB/10.0 GiB)
[1203830.268093] sd 4:0:0:1: [sdh] 4096-byte physical blocks
[1203830.268847] sd 4:0:0:1: [sdh] Write Protect is off
[1203830.268858] sd 4:0:0:1: [sdh] Mode Sense: 8f 00 00 08
[1203830.269033] sd 4:0:0:1: [sdh] Write cache: disabled, read cache: enabled,
doesn't support DPO or FUA
[snip]
```

5.5 /proc/scsi/scsi 파일

/proc/scsi/scsi 파일은 Linux 호스트의 볼륨 및 대상에 대한 추가 정보를 제공할 수 있습니다.

```
# cat /proc/scsi/scsi
Attached devices:
[snip]
Host: scsi1 Channel: 00 Id: 00 Lun: 01
  Vendor: COMPELNT Model: Compellent Vol   Rev: 0605
  Type:   Direct-Access                    ANSI  SCSI revision: 05
Host: scsi1 Channel: 00 Id: 02 Lun: 01
  Vendor: COMPELNT Model: Compellent Vol   Rev: 0605
  Type:   Direct-Access                    ANSI  SCSI revision: 05
Host: scsi1 Channel: 00 Id: 03 Lun: 01
  Vendor: COMPELNT Model: Compellent Vol   Rev: 0605
  Type:   Direct-Access                    ANSI  SCSI revision: 05
Host: scsi1 Channel: 00 Id: 03 Lun: 02
  Vendor: COMPELNT Model: Compellent Vol   Rev: 0605
  Type:   Direct-Access                    ANSI  SCSI revision: 05
[snip]
```



6 Dell Compellent Command Utility

Storage Center SAN 은 Dell Compellent Command Utility(CompCU)라는 원격 명령 유틸리티를 통해 일상적인 기능을 관리할 수 있습니다. 이를 통해 Linux 운영 체제와 Storage Center 간에 SAN 작업의 스크립팅 및 자동화 통합이 가능합니다. CompCU 는 Java 패키지 응용 프로그램이며 Linux 호스트에 Java 설치가 필요합니다. CompCU 는 시간을 크게 절약할 수 있도록 일반적인 관리 작업을 스크립트로 작성하고 Storage Center 볼륨 및 Replays 를 관리하기 위한 일관된 프레임워크를 제공합니다.

CompCU 를 사용하려면 호스트에 적절한 Java 릴리스가 설치되어 있어야 합니다. 자세한 내용은 명령 유틸리티 사용 설명서(부록 A)를 참조하십시오. CompCU.jar 개체는 Dell Compellent 지원 사이트에서 다운로드할 수 있습니다. Linux 호스트에 설치되면 이 도구를 사용하여 셸 프롬프트에서 Storage Center 작업을 수행할 수 있습니다. 이 작업은 새로운 또는 기존 사용자 관리 스크립트에 통합할 수 있습니다. CompCU 의 일반적인 사용 사례는 다음과 같습니다.

- 볼륨 생성, 서버에 매핑.
- Replays 수행, Replay 복구 등.

아래의 예는 CompCU 의 유용성에 대한 포괄적인 내용을 다루지는 않습니다. CompCU 로 자동화될 수 있는 작업 유형에 대한 초기 통찰력을 제공하도록 설계되었습니다.

6.1 Java 확인, CompCU 기능 구성 및 테스트

먼저 Linux 호스트에 Java(RTE v1.6.x 이상)를 설치합니다. Java 런타임은 운영 체제와 함께 이미 설치되어 있으며 아래 표시된 명령으로 확인할 수 있습니다.

```
# /usr/bin/java -version
java version "1.7.0_07"
Java(TM) SE Runtime Environment (build 1.7.0_07-b10)
Java HotSpot(TM) Server VM (build 23.3-b01, mixed mode)
```

Dell Compellent 지원

사이트(http://kc.compellent.com/Published%20Documents/CU060401_001.zip)에서 CompCU 패키지를 다운로드합니다. 패키지에는 PDF 사용 설명서와 필수 CompCU.jar 파일이 포함됩니다. 이 CompCU.jar 파일을 논리 파일 시스템 위치에 저장하십시오. 도움말 및 사용법 구문을 표시하려면 아래 명령을 실행하여 CompCU 가 Java 에서 작동하는지 확인하십시오.

```
# /usr/bin/java -jar ./CompCU.jar -h
Compellent Command Utility (CompCU) 6.4.1.1
```

```
usage: java -jar CompCU.jar [Options] "<Command>"
  -c <arg>                Run a single command (option must be within
                           quotes)
  -default                Saves host, user, and password to encrypted file
  -defaultname <arg>      File name to save default host, user, and password
                           encrypted file to
```



```

-file <arg>          Save output to a file
-h                  Show help message
-host <arg>         IP Address/Host Name of Storage Center Management
                    IP
-password <arg>     Password of user
-s <arg>            Run a script of commands
-user <arg>         User to log into Storage Center with
-verbose           Show debug output
-xmloutputfile <arg> File name to save the CompCU return code in xml
                    format. Default is cu_output.xml.

```

[snip]

CompCU 를 사용하여 쉽게 액세스할 수 있도록 다음과 같이 암호화된 암호 파일을 구성하기 위해 초기에 - default 스위치를 사용하여 이 도구를 실행할 수 있습니다. default.cli 라는 파일이 로컬 디렉터리에 작성됩니다. 이 파일은 명확성과 사용을 위해 필요에 따라 이름이 바뀔 수 있습니다

```

# /usr/bin/java -jar ./CompCU.jar -default -host 172.16.2.109 -user Admin -
password XXX
Compellent Command Utility (CompCU) 6.4.1.1

```

```

=====
User Name:          Admin
Host/IP Address:    172.16.2.109
=====

```

```

Connecting to Storage Center: 172.16.2.109 with user: Admin
java.lang.IllegalStateException: TrustManagerFactoryImpl is not initialized
Saving CompCu Defaults to file [default.cli]...
The "default.cli" file may then be referenced in other commands to login to the
same Storage Center and perform tasks. A separate .cli file may be created for
each Storage Center under management with each containing the appropriate login
credentials for the respective Storage Center array. The example below
demonstrates a "volume show" command applied to the Storage Center located at IP
address 172.16.2.109.

```

```

# /usr/bin/java -jar ./CompCU.jar -defaultname default.cli -host 172.16.2.109 -
user Admin -password XXX -c "volume show"
Compellent Command Utility (CompCU) 6.4.1.1

```

```

=====
User Name:          Admin
Host/IP Address:    172.16.2.109
Single Command:     volume show
=====

```

```

Connecting to Storage Center: 172.16.2.109 with user: Admin
java.lang.IllegalStateException: TrustManagerFactoryImpl is not initialized
Running Command: volume show

```



Index	Name	Status	ConfigSize	ActiveSize
ReplaySize	Folder			
StorageProfile	DeviceI	D		
SerialNumber	ConfigSizeBlock	ActiveSizeBlock	ReplaySizeBlock	
MaxWriteSizeBlo	ReadCache	WriteCache		
283	Fal-asm-mirror-test-failgroup1	Up	100.00 GB	31.33 GB
0.00 KB	Oracle/11gR2/ASM-Mirror			
Recommended	6000d31		00000650000000000000000012f	
00000065-0000012f	209715200	65712128	0	0
Enabled	Enabled			
290	Fal-asm-mirror-sp-fg1	Up	100.00 GB	7.94 GB
0.00 KB	Oracle/11gR2/ASM-Mirror			
Recommended	6000d31		000006500000000000000000136	
00000065-00000136	209715200	16658432	0	0
Enabled	Enabled			
824	ibmsvc00-managed-mdisk1	Up	500.00 GB	98.02 GB
0.00 KB	IBMSVC			
Recommended	6000d31		00000650000000000000000034c	
00000065-0000034c	1048576000	205561856	0	0
Enabled	Enabled			

[snip]

Successfully finished running Compellent Command Utility (CompCU) application.

6.2 CompCU 를 사용하여 일반적인 작업 자동화

이 섹션에서는 Linux 에서 CompCU 를 사용하여 Storage Center 작업을 관리하기 위한 몇 가지 사용 사례를 설명합니다. 위에서 언급했듯이 이 예제는 CompCU 를 사용하는 Linux 셸 프롬프트에서 쉽게 수행할 수 있는 작업 유형을 나타냅니다. 이 예제는 시스템 관리자에게 이 강력한 도구 세트를 숙지시키는 시작점으로만 의미가 있습니다.

6.2.1 CompCU 를 사용하여 단일 볼륨 생성

이 예제는 CompCU 를 사용하여 Linux 호스트에서 hank_100g_00 이라는 단일 100GB Storage Center 볼륨을 생성한 다음, Storage Center Linux 폴더에 배치하는 방법을 보여줍니다. 볼륨은 Linux hank 호스트에 매핑됩니다.

```
# /usr/bin/java -jar ./CompCU.jar -defaultname default.cli -host 172.16.2.109 -
user Admin -password XXX -c "volume create -name hank_100g_00 -folder Linux -
server hank -size 100g"
```



6.2.2 CompCU 를 사용하여 Replay 및 Replay View 생성

이 예제는 하나의 CompCU 명령을 사용하여 보여줍니다.

- Storage Center 에서 기존 hank_100g_00 볼륨의 Replay, hank_100g_00_Replay 생성
- 이 언급된 Replay 에서 Replay View, hank_100g_00_View 생성 및
- Replay View 를 Linux 호스트, dean 에 매핑.

```
# /usr/bin/java -jar ./CompCU.jar -defaultname default.cli -host 172.16.2.109 -  
user Admin -password XXX -c "replay create -volume 'hank_100g_00' -name  
'hank_100g_00_Replay' -view 'hank_100g_00_RpView' -server 'dean'"
```

6.2.3 CompCU 를 사용한 다중 볼륨의 신속 배포

이 마지막 예제에서는 CompCU 를 사용하여 Storage Center 에서 빠르게 볼륨을 생성 및 배포하고 이 볼륨을 Linux 호스트 dean 에 매핑하는 방법을 보여줍니다.

```
# for i in 0 1 2 3 4 5 6 7 8 9; do /usr/bin/java -jar ./CompCU.jar -defaultname  
default.cli -host 172.16.2.109 -user Admin -password XXX -c "volume create -name  
dean_10g_0${i} -folder Linux -server 'dean' -size 10g"; done
```



A 추가 자료

Dell Storage Center System Manager 6.5 관리자 안내서

<http://kc.compellent.com/Knowledge%20Center%20Documents/680-019-016.pdf>

Dell Enterprise Manager 2014 R2 관리자 안내서

<http://kc.compellent.com/Knowledge%20Center%20Documents/680-017-021.pdf>

SC8000 Connectivity 안내서

<http://kc.compellent.com/Knowledge%20Center%20Documents/680-027-013.pdf>

Dell Storage Center 6.0 Command Utility(CompCU) 참조 안내서

<http://kc.compellent.com/Knowledge%20Center%20Documents/680-018-007.pdf>

Red Hat Enterprise Linux 7.0 베타 릴리스 정보

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/7-Beta/html/7.0_Release_Notes/index.html

Red Hat Enterprise Linux 7.0 베타 시스템 관리자 안내서

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/7-Beta/html/System_Administrators_Guide/index.html

Red Hat Enterprise Linux 문서 포털

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/

Red Hat Labs(베타)

<https://access.redhat.com/labs/>



B Configuration details(구성 세부 정보)

구성 요소	설명
운영 체제	Red Hat Enterprise Linux 7.1 GA(Maipo)
드라이버 버전	드라이버 버전 = 8.04.00.04.06.3-k-debug BIOS 버전 = 3.00
펌웨어 버전	펌웨어 버전 = 5.06.05(90d5)
응용 프로그램	NA
케이블 연결	QLE2562 8Gb Dual Port PCIe FC HBA Dell 12Gbps SAS HBA
서버	Dell R630 x 2
보관 시	Dell Storage Center SC 8000 SCOS 6.6.x 가상 포트 모드 Dell Storage Center SC v2x00 SCOS 6.6.x 가상 포트 모드
스위치	Dell PowerConnect 5500 시리즈

