KVM/DR

백서 | 크로스 하이퍼바이저 DR 시리즈 | 2 부 (전 2 부)

KVMDR 의 크로스 하이퍼바이저 DR 구현 방식

Proxmox VE, oVirt, OLVM, RHV 전반의 일관된 복구 경험

KVMDR AI Ltd | kvmdr.ai | 2026 년 11 월

요약

1 부에서는 Proxmox VE 와 oVirt 가 같은 KVM 가상 머신을 공유하지만 관리 방식은 매우 다르다는 점을 살펴보았습니다. 이 백서는 KVMDR 이 그 차이를 어떻게 메우는지 설명합니다. 플랫폼 중립적인 복제본, 양 끝의 플랫폼별 어댑터, 그리고 랜섬웨어로부터의 특정 시점 복구, 운영 중단 없는 테스트, 유연한 페일백 등 엔터프라이즈 DR 에 기대되는 복구 기능입니다.

1. 설계 원칙: 가운데는 중립적으로

대부분의 DR 도구는 하나의 하이퍼바이저를 중심으로 만들어지고 다른 하이퍼바이저는 나중에 덧붙여집니다. KVMDR 은 처음부터 KVM 을 위해 설계되었으며, 보호 대상 데이터를 어느 플랫폼에도 속하지 않는 형태로 보관합니다. 플랫폼에 대한 지식은 양 끝의 어댑터에만 있으며, 어댑터는 소스에서 읽는 방법과 대상에서 VM 을 구축하는 방법을 알고 있습니다.

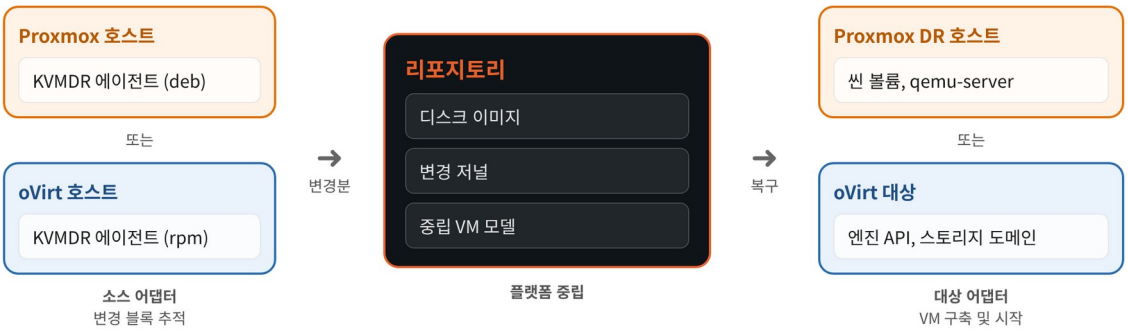


그림 1. KVMDR 크로스 하이퍼바이저 데이터 경로

- **소스 어댑터.** Proxmox(deb)와 oVirt(rpm) 모두를 위해 패키징된 KVMDR 호스트 에이전트가 각 VM 디스크의 변경 블록을 추적하여 리포지토리로 전송합니다. 게스트 내부에는 아무것도 설치하지 않습니다.
- **플랫폼 중립 리포지토리.** 디스크는 변경 저널과 함께 일반 이미지로 보관되며, CPU, 메모리, 디스크, NIC, 부팅 설정 등 VM 에 대한 중립적인 기술 정보도 함께 보관됩니다.
- **대상 어댑터.** Proxmox 에서는 KVMDR 이 DR 호스트에 직접 VM 을 구축합니다. oVirt 에서는 선택한 스토리지 도메인에 엔진을 통해 VM 과 디스크를 등록합니다.

KVMDR 이 각 스택에 연결되는 방식

1 부에서 살펴본 것처럼 두 플랫폼은 모두 QEMU 에서 끝나지만 QEMU 위는 모두 다릅니다. KVMDR 은 두 플랫폼에 공통인 QEMU 계층에서 변경을 캡처하고, 대상에서 VM 을 구축하고 시작할 때처럼 꼭 필요한 경우에만 각 플랫폼의 관리 계층과 통신합니다.



그림 2. QEMU 계층에서의 변경 캡처

1. 초기 동기화: KVMDR 은 각 디스크에 더티 비트맵을 활성화해 그 순간부터의 모든 쓰기를 기록한 뒤, 읽기 전용 NBD 익스포트를 통해 디스크 전체를 고정 크기 청크 단위로 읽습니다. 동기화가 중단되면 마지막으로 완료된 청크부터 재개합니다.
2. 지속적 복제: KVMDR 은 비트맵에 표시된 블록만 읽어 리포지토리 저널에 쓰고, 새 비트맵을 시작합니다. 전체 복사를 시작하기 전에 비트맵을 활성화하므로 둘 사이의 틈으로 쓰기가 누락되지 않습니다.
3. 복구: 대상 어댑터가 리포지토리의 이미지와 저널로부터 선택한 시점의 디스크를 만들어 대상 플랫폼에 전달합니다.

연동 지점	Proxmox VE	oVirt, OLVM, RHV
QEMU 와의 통신	Proxmox 자체가 사용하는 QMP 소켓을 통해 직접	각 호스트에서 vdsm 및 libvirt 와 함께 동작
VM 설정 읽기	/etc/pve 의 VM 구성	엔진 REST API
DR 디스크 생성	DR 호스트 스토리지의 씬 볼륨	엔진을 통해 대상 스토리지 도메인에 디스크 이미지 생성
VM 생성 및 시작	DR 호스트의 qemu-server	엔진 REST API (vdsm 과 libvirt 에 지시)
컨테이너	LVM thin 볼륨의 블록 필터	해당 없음

> **QEMU 계층에서 캡처하는 이유**
> QEMU 더티 비트맵과 NBD 익스포트는 표준 QEMU 기능이며, Proxmox Backup Server 와 oVirt 증분 백업도 같은 기능을 사용합니다. 이 계층에서 동작하므로 KVMDR 은 두 플랫폼에 하나의 캡처 방식을 사용하며, 게스트 내부 소프트웨어가 필요 없고, 복제 자체는 어느 관리 계층에도 의존하지 않습니다.

2. 크로스 하이퍼바이저 복구 진행 방식

1. 운영자가 VM 또는 일관성 그룹과 복구 시점(최신 시점 또는 랜섬웨어 발생 이전의 깨끗한 시점)을 선택합니다.
2. KVMDR 이 대상의 준비 상태(스토리지, 네트워크 매핑, oVirt 의 경우 엔진)를 확인합니다.
3. 대상 어댑터가 중립 모델로부터 VM 을 구축하고, 해당 플랫폼에 맞게 이름과 네트워크를 변환합니다.
4. 선택한 시점의 디스크가 제공됩니다.
5. 네트워크 규칙이 적용됩니다. 새 IP 주소를 할당(Re IP)하거나 원래 MAC 주소를 유지합니다.
6. 일관성 그룹의 VM 은 같은 시점에서 함께 시작됩니다.

> **어느 방향이든 같은 절차**
> Proxmox 에서 oVirt 로, oVirt 에서 Proxmox 로 모두 같은 워크플로와 같은 화면을 사용합니다. 1 부에서 설명한 플랫폼 차이는 어댑터가 흡수합니다.

3. 차이는 KVMDR 이 처리

플랫폼 차이 (1 부)	KVMDR 의 처리 방식
oVirt 는 엔진이 필요	엔진 API 로 VM 을 생성하고 시작하며, 복구 과정에서 엔진 상태를 확인
디스크는 스토리지 도메인에 위치	크로스 하이퍼바이저 프로필에서 대상 클러스터와 스토리지 도메인을 선택
네트워크 모델의 차이	소스 브리지에서 대상 네트워크로의 네트워크 맵. Re IP 또는 MAC 유지 지원
명명 규칙의 차이	이름 자동 변환 (예: Proxmox 에서는 밑줄을 하이픈으로)
컨테이너는 Proxmox 에서만	컨테이너는 VM 과 같은 기능으로 Proxmox 에서 Proxmox 로 보호
테스트 방식의 차이	시뮬레이션이 플랫폼에 맞게 동작: Proxmox 에서는 공간 효율적, oVirt 에서는 베이스라인 사용

4. DR 사이트 용량 산정

1 부에서는 oVirt 대상의 DR 테스트가 Proxmox 보다 더 많은 공간을 필요로 한다고 언급했습니다. 그 이유는 테스트를 위해 VM 의 임시 쓰기 가능 사본을 만드는 방식이 플랫폼마다 다르기 때문입니다.

Proxmox 에 추가 공간이 거의 필요 없는 이유

Proxmox 대상에서 DR 디스크는 KVMDR 이 직접 제어하는 DR 호스트의 씬 볼륨입니다. 시뮬레이션 시 KVMDR 은 즉시 씬 스냅샷을 만들고 그로부터 테스트 VM 을 부팅합니다. 새 공간을 차지하는 것은 테스트 VM 이 쓴 블록뿐이며, 테스트가 끝나면 폐기됩니다. 그동안에도 복제는 원래 디스크로 계속 진행됩니다.

oVirt 에 테스트 베이스라인이 필요한 이유

oVirt 대상에서 DR 디스크는 스토리지 도메인 안의 이미지이며, 스냅샷이나 클론은 엔진만 만들 수 있습니다. 1 부에서 살펴본 것처럼 oVirt 스냅샷은 디스크의 볼륨 체인에 qcow2 오버레이를 추가하고 쓰기를 그쪽으로 전환합니다. 그 체인은 엔진 데이터베이스에서 하나의 VM 에 속합니다. KVMDR 이 테스트를 위해 DR 디스크의 스냅샷을 만들면 복제와 테스트 VM 이 같은 체인을 두고 경쟁하게 됩니다. 엔진을 통해 디스크를 독립된 새 이미지로 클론하면 전체 복사가 됩니다.

따라서 KVMDR 은 스토리지 도메인에 별도의 테스트 베이스라인을 유지합니다. 테스트 VM 은 이 독립 사본에서 부팅하고, 그동안 복제는 DR 디스크에 영향을 주지 않고 계속됩니다. 베이스라인의 크기는 보호 대상 디스크와 같습니다.

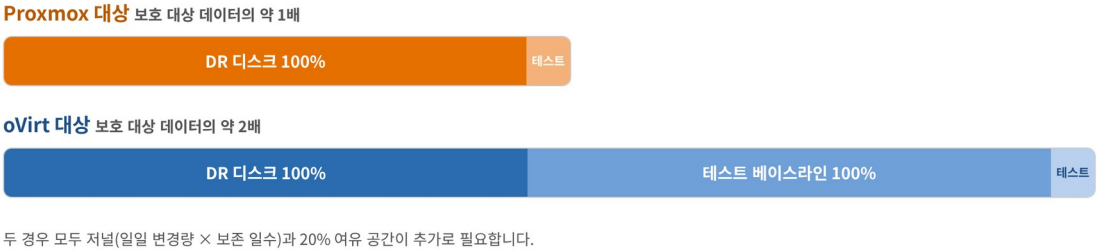


그림 3. 시뮬레이션 중 DR 대상 공간

용량 산정 가이드

항목	Proxmox 대상	oVirt 대상
DR 디스크	보호 대상 데이터의 1 배	보호 대상 데이터의 1 배
테스트 베이스라인	없음	보호 대상 데이터의 1 배
테스트 중 쓰기	소량, 임시	소량, 임시
복구 이력 (저널)	일일 변경량 × 보존 일수	일일 변경량 × 보존 일수
여유 공간	20% 이상	20% 이상

> **예: 보호 대상 VM 2 TB**
> Proxmox 대상: DR 디스크 2 TB + 저널 + 여유 공간 20%. oVirt 대상: DR 디스크 2 TB + 테스트 베이스라인 2 TB + 저널 + 여유 공간 20%. oVirt 에서는 DR 디스크 공간의 약 2 배를 계획하십시오. 저널 크기는 하루에 변경되는 데이터 양과 이력 보존 기간에 따라 달라집니다.

5. 플랫폼 전반의 엔터프라이즈 DR 기능

- **준동기 복제.** 변경 사항을 지속적으로 캡처하여 최소 1 초 단위의 복구 지점을 제공합니다.
- **특정 시점 복구.** 어젯밤 백업이 아니라 랜섬웨어 공격 직전 시점으로 되돌릴 수 있습니다.
- **AI 랜섬웨어 탐지.** 비정상적인 쓰기 패턴을 발생 즉시 탐지하고, 마지막 깨끗한 시점을 표시합니다.
- **시뮬레이션.** 보호를 계속 유지하면서 격리된 네트워크에서 전체 복구 테스트를 수행하며, 운영 환경에 영향을 주지 않습니다.
- **일관성 그룹.** 애플리케이션과 데이터베이스처럼 연결된 VM 을 같은 시점으로 복구합니다.
- **유연한 페일백.** 페일오버 이후의 변경분만 전송해 원래 사이트로 돌아가거나, 필요한 경우 VM 또는 컨테이너 전체를 복구할 수 있습니다.
- **고가용성 관리.** 액티브와 스탠바이 매니저 쌍으로 사이트를 잃어도 DR 제어가 유지됩니다.
- **게스트 내 에이전트 불필요.** 보호는 하이퍼바이저 호스트에서 실행됩니다. 게스트에는 파일 복원을 위한 표준 QEMU 게스트 에이전트만 있으면 됩니다.

6. KVMDR 이 뛰어난 이유

- **VMware 에서 가져온 것이 아닌, KVM 을 위해 설계.** KVMDR 은 처음부터 KVM 과 QEMU 를 중심으로 설계되었기 때문에 두 플랫폼이 하나의 중립 모델에 깔끔하게 대응됩니다.
- **두 플랫폼, 양방향, 하나의 제품.** 하나의 라이선스와 하나의 워크플로로 Proxmox VE, oVirt, OLVM, RHV 를 소스와 대상 모두로 지원합니다.
- **단순 복사가 아닌 복구.** 특정 시점 복구, Re IP, 일관성 그룹, 시뮬레이션으로 실제 재해 상황에서 사용할 수 있는 복제본을 만듭니다.
- **랜섬웨어 대응.** 탐지와 깨끗한 시점으로의 복구는 별도 제품이 아니라 복제 경로의 일부입니다.
- **엔터프라이즈 DR 경험.** KVMDR 팀은 엔터프라이즈 재해 복구 솔루션의 구축과 판매에서 수십 년의 경험을 갖고 있으며, 그 노하우를 KVM 에 적용합니다.

7. 대표적인 활용 사례

- **혼합 환경.** 메인 데이터센터에는 oVirt 또는 OLVM, 보조 사이트에는 Proxmox 를 두고 서로를 보호합니다.
- **플랫폼 이전.** oVirt 나 RHV 에서 Proxmox 로 이전하는 팀은 복제하고, 시뮬레이션으로 테스트한 뒤 전환할 수 있으며, 소스는 롤백용으로 유지됩니다.
- **저비용 DR 사이트.** oVirt 또는 OLVM 운영 환경을 범용 하드웨어 기반의 Proxmox DR 사이트로 보호합니다.

8. 결론

Proxmox VE 와 oVirt 의 관리 방식은 매우 다르지만, 실행하는 가상 머신은 같습니다. KVMDR 은 이 공통 기반을 활용합니다. 보호 대상 데이터를 중립적으로 유지하고, 플랫폼 지식은 양 끝의 어댑터에 두며, 이를 엔터프라이즈 DR 기능으로 감쌉니다. 그 결과 모든 KVM 플랫폼에서 어느 방향으로든 일관된 복구 경험을 제공합니다.

직접 확인해 보십시오. kvmdr.ai 에서 7 일 무료 체험을 시작하거나 데모를 예약할 수 있습니다.

KVMDR 소개

KVMDR AI Ltd 는 KVM 기반 하이퍼바이저(Proxmox VE, oVirt, Oracle Linux Virtualization Manager, Red Hat Virtualization)를 위한 재해 복구(DR) 솔루션을 개발하는 영국 기업입니다. KVMDR 은 준동기 복제, 랜섬웨어로부터의 특정 시점 복구, 운영 중단 없는 DR 테스트, AI 기반 위협 탐지를 하나의 플랫폼으로 제공합니다. 7 일 무료 체험은 kvmdr.ai 에서 시작할 수 있습니다.