KVM/DR

백서 | 크로스 하이퍼바이저 DR 시리즈 | 1 부 (전 2 부)

Proxmox VE 와 oVirt: 같은 엔진, 다른 세계

크로스 하이퍼바이저 재해 복구로 무엇을 할 수 있고 무엇을 할 수 없는지, 그리고 플랫폼마다 복구 방식이 다른 이유

KVMDR AI Ltd | kvmdr.ai | 2026 년 11 월

요약

Proxmox VE 와 oVirt 는 모두 KVM 과 QEMU 에서 가상 머신을 실행하므로, VM 은 데이터를 그대로 유지한 채 두 플랫폼 사이를 이동할 수 있습니다. 그러나 두 플랫폼의 관리 방식은 크게 다르며, 이 차이가 각 환경에서 재해 복구로 달성할 수 있는 범위를 결정합니다. 이 백서는 두 아키텍처를 비교하고, 방향별로 가능한 것을 정리하며, oVirt 를 대상으로 한 복구가 Proxmox 로의 복구보다 더 많은 계획을 필요로 하는 이유를 설명하고, 실무적인 모범 사례 체크리스트를 제시합니다. 2 부에서는 KVMDR 이 크로스 하이퍼바이저 DR 을 어떻게 구현하는지 설명합니다.

1. 크로스 하이퍼바이저 DR 이 중요한 이유

오늘날 많은 조직이 둘 이상의 KVM 플랫폼을 운영합니다. 일부는 oVirt 나 Red Hat Virtualization 에서 Proxmox VE 로 이전하고 있고, 다른 조직은 데이터센터에서 oVirt 나 Oracle Linux Virtualization Manager 를 유지하면서 엣지나 보조 사이트에서 Proxmox 를 사용합니다. 어느 경우든 같은 질문이 생깁니다. 한 플랫폼이 다른 플랫폼을 보호할 수 있는가?

답은 '그렇다'입니다. 가상 머신 자체는 양쪽에서 동일한, QEMU 장치와 VirtIO 드라이버를 갖춘 KVM 게스트이기 때문입니다. 다른 것은 VM 을 둘러싼 모든 요소이며, 바로 그 부분에서 계획이 중요해집니다.

> **핵심 요약**
>
> Proxmox 는 각 호스트가 스스로를 관리하는 호스트들의 집합입니다. oVirt 는 중앙 엔진이 호스트를 관리합니다. Proxmox 로의 복구는 DR 호스트에만 의존합니다. oVirt 로의 복구는 oVirt 엔진이 실행 중인지에도 의존합니다.

2. 두 가지 아키텍처

Proxmox VE: 각 호스트가 스스로를 관리

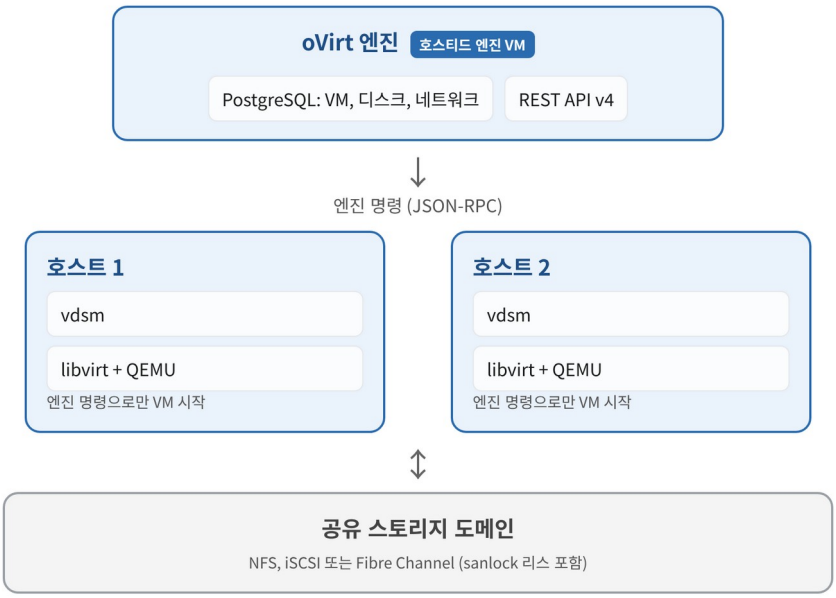
각 Proxmox 노드는 자체 API, 웹 인터페이스, 명령줄 도구를 갖습니다. 노드들은 구성을 공유하는 클러스터를 구성할 수 있지만 중앙 관리자는 없습니다. 어떤 노드든 LVM thin, ZFS, 디렉터리, Ceph 등 자신이 접근할 수 있는 스토리지를 사용해 가상 머신과 컨테이너를 단독으로 생성, 시작, 중지할 수 있습니다.



그림 1. Proxmox VE: 각 노드가 스스로를 관리

oVirt, OLVM, RHV: 중앙 엔진이 모든 것을 제어

oVirt 에서는 엔진이 모든 VM, 디스크, 네트워크, 스토리지 도메인의 데이터베이스를 보유합니다. 호스트는 vdsm 서비스를 실행하며 엔진이 지시할 때만 VM 을 시작합니다. 디스크는 NFS, iSCSI, Fibre Channel 위의 공유 스토리지 도메인 안에 있는 이미지입니다. 엔진 자체는 보통 같은 호스트에서 호스티드 엔진 VM 으로 실행됩니다.

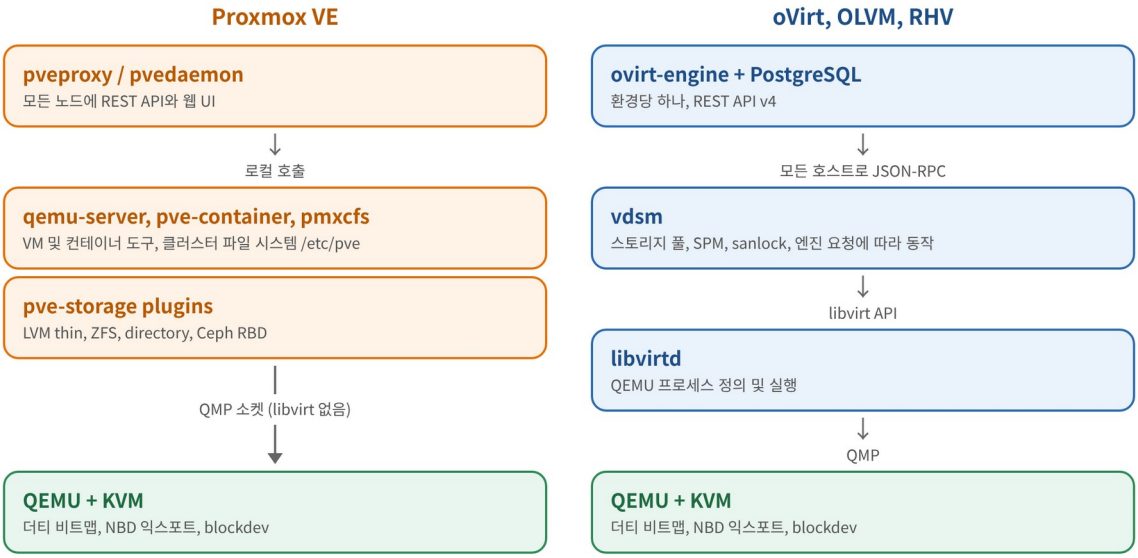


VM은 엔진 데이터베이스에만 존재합니다. 엔진이 없으면 새 VM도 없습니다.

그림 2. oVirt, OLVM, RHV: 중앙 엔진이 호스트를 관리

소프트웨어 스택의 내부

두 플랫폼 모두 같은 곳, 즉 KVM 게스트를 실행하는 QEMU 프로세스에서 끝납니다. 차이는 QEMU 위의 계층에서 생기며, 이 계층이 복제, 스냅샷, 복구의 동작 방식을 결정합니다.



최하위 계층은 같고, 그 위의 제어가 다릅니다. Proxmox는 QEMU를 직접 제어하고, oVirt는 엔진, vdsm, libvirt, QEMU 순으로 거칩니다.

그림 3. 소프트웨어 스택 (위에서 아래로)

계층	Proxmox VE	oVirt, OLVM, RHV
관리 API	모든 노드의 pveproxy 와 pvedaemon	ovirt-engine (환경당 하나)

계층	Proxmox VE	oVirt, OLVM, RHV
VM 정의	/etc/pve/qemu-server 의 텍스트 파일. pmxcfs 가 corosync 로 복제	엔진 PostgreSQL 데이터베이스의 행
호스트 에이전트	qemu-server 와 pve-container (각 노드의 일부)	vdsm (엔진 요청에만 동작)
QEMU 제어	QMP 소켓을 통해 직접. Proxmox 는 libvirt 를 사용하지 않음	vdsm 이 제어하는 libvirt 를 통해
스토리지 계층	pve-storage 플러그인: LVM thin, ZFS, 디렉터리, Ceph RBD	스토리지 도메인으로 구성된 스토리지 풀. SPM 호스트가 조정
잠금	pmxcfs 클러스터 잠금	공유 스토리지의 sanlock 리스

스토리지의 내부 동작

Proxmox 는 각 VM 디스크를 스토리지 플러그인에 맡깁니다. LVM thin 에서는 디스크가 device mapper 씬 볼륨이고, ZFS 에서는 zvol 입니다. 스냅샷은 스토리지 자체(dm thin 스냅샷 또는 ZFS 스냅샷)가 거의 즉시 생성하며, 단독으로 사용할 수 있는 별도의 블록 장치가 됩니다.

oVirt 는 스토리지를 스토리지 도메인으로 구성된 스토리지 풀로 묶습니다. 파일 도메인(NFS)에서 각 디스크는 하나 이상의 볼륨 파일을 담은 이미지 디렉터리이며, 각 볼륨에는 메타데이터 파일과 sanlock 리스가 있습니다. 블록 도메인(iSCSI, FC)에서는 각 스토리지 도메인이 LVM 볼륨 그룹이고 각 볼륨은 논리 볼륨입니다. oVirt 의 스냅샷은 기존 볼륨 위에 새 qcow2 볼륨을 추가해 체인을 형성합니다. 볼륨 생성은 공유 메타데이터를 변경하므로, 엔진의 지시에 따라 선출된 하나의 호스트인 스토리지 풀 매니저(SPM)가 수행합니다.



그림 4. 스냅샷이 하나 있는 디스크

이 차이는 재해 복구에 중요합니다. Proxmox 에서 스냅샷은 독립된 장치이므로 DR 도구가 원본을 건드리지 않고 열고, 부팅하고, 삭제할 수 있습니다. oVirt 에서 스냅샷은 VM 이 쓰는 볼륨을 바꾸며, 그 체인은 엔진 데이터베이스에서 하나의 VM 에 속합니다.

엔진 자체의 위치

대부분의 oVirt 환경에서 엔진은 같은 호스트에서 호스티드 엔진 VM 으로 실행됩니다. 각 호스트의 두 서비스인 ovirt-ha-agent 와 ovirt-ha-broker 가 공유 스토리지의 sanlock 락스페이스를 사용해

엔진을 계속 실행합니다. broker 가 중지되거나 락스페이스를 획득하지 못하면 호스티드 엔진은 중지된 상태로 남고, 복구된 VM 의 시작을 포함한 모든 엔진 작업도 함께 멈춥니다.

항목	Proxmox VE	oVirt, OLVM, RHV
관리	각 노드가 자체 API 와 UI 보유	모든 호스트를 하나의 중앙 엔진이 관리
VM 이 정의되는 곳	노드의 구성 파일	엔진 데이터베이스
VM 을 시작하는 주체	어떤 노드든 단독으로	엔진 (vdsm 을 통해)
디스크 위치	노드 스토리지의 볼륨	공유 스토리지 도메인의 이미지
컨테이너	지원 (LXC)	미지원
VM 이름	유효한 호스트 이름이어야 함. 밑줄 불가	밑줄 허용
보호해야 할 구성 요소	없음. 각 노드가 독립적	엔진

3. 방향별로 가능한 것

기능	Proxmox 에서 oVirt 로	oVirt 에서 Proxmox 로
VM 복제	가능	가능
컨테이너 (LXC)	불가. oVirt 에는 컨테이너가 없으므로 Proxmox 에서 Proxmox 로 보호	해당 없음
대상 스토리지	oVirt 파일 스토리지 도메인 (NFS)	DR 호스트의 씬 프로비저닝 스토리지
운영 중단 없는 DR 테스트	가능. 테스트 베이스라인용 추가 공간 필요	가능. 공간 효율적
페일오버	가능. oVirt 엔진 실행 필요	가능. DR 호스트만 필요
페일오버 시 IP 변경 (Re IP)	가능	가능
대상의 VM 이름	유지	밑줄은 하이픈으로 변환
실행 중인 VM 으로 파일 복원	가능 (QEMU 게스트 에이전트 필요)	가능 (QEMU 게스트 에이전트 필요)

현재 제한 사항

- **컨테이너는 Proxmox 에서만.** oVirt 는 컨테이너를 지원하지 않으므로 LXC 컨테이너는 Proxmox 에서 Proxmox 로 보호합니다.
- **oVirt 에서는 파일 스토리지 도메인만.** iSCSI 및 Fibre Channel 스토리지 도메인은 아직 대상으로 지원되지 않습니다.
- **엔진이 없으면 oVirt 로 페일오버 불가.** oVirt 호스트는 엔진 없이 새 VM 을 시작할 수 없습니다.

4. oVirt 대상에서 복구가 다른 이유

엔진이 실행 중이어야 함

Proxmox 에서는 복구가 DR 호스트와 직접 통신하며, DR 호스트는 스스로 VM 을 생성하고 시작할 수 있습니다. oVirt 에서는 모든 VM 과 디스크를 엔진에 등록해야 하며, VM 을 시작할 수 있는 것도 엔진뿐입니다. 호스티드 엔진이 중지되었거나 비정상이면 복구는 대기합니다. 이것이 두 대상 간의 가장 큰 차이입니다.

디스크는 스토리지 도메인에 있어야 함

Proxmox 에서 DR 디스크는 DR 호스트에 있으며 빠르게 생성하고 교체할 수 있습니다. oVirt 에서 디스크는 스토리지 도메인 안의 이미지이며 엔진을 통해 생성됩니다. 대상 스토리지 도메인은 미리 선택하고 보호 대상 데이터에 맞게 용량을 산정해야 합니다.

oVirt 에서는 테스트에 더 많은 공간이 필요

Proxmox 에서 DR 테스트는 추가 공간이 거의 필요하지 않습니다. oVirt 에서는 테스트 베이스라인 사본을 유지하므로, 스토리지 도메인에 보호 대상 데이터와 그 베이스라인을 함께 담을 수 있어야 합니다. 그 이유와 대상별 용량 산정 방법은 이 시리즈의 2 부에서 설명합니다.

이름과 네트워크는 변환됨

Proxmox VM 이름에는 밑줄을 사용할 수 없습니다. 네트워크 모델도 다릅니다. Proxmox 는 vmbr0 같은 Linux 브리지를, oVirt 는 논리 네트워크와 vNIC 프로필을 사용합니다. 모든 소스 네트워크에는 명확한 대상 네트워크가 필요합니다.

5. 모범 사례 체크리스트

보호 시작 전

1. VM 마다 어느 플랫폼을 기본으로 할지 정하고 일관되게 유지합니다.
2. oVirt 가 대상이라면 엔진을 최소 두 대의 호스트에서 고가용성 호스티드 엔진으로 실행합니다.
3. oVirt 가 대상이라면 보호 대상 데이터와 테스트 베이스라인을 담을 수 있는 크기의 NFS 데이터 도메인을 만듭니다.
4. 컨테이너는 Proxmox 에서 Proxmox 로 보호합니다.
5. VirtIO 디스크와 NIC 를 사용하고 모든 VM 에 QEMU 게스트 에이전트를 설치합니다.
6. Proxmox 에서 실행될 수 있는 VM 의 이름에는 밑줄을 쓰지 않습니다.

네트워크

1. 모든 소스 브리지 또는 네트워크를 대상 네트워크에 매핑합니다.
2. 페일오버 시 IP 주소를 변경할지 유지할지 VM 별로 결정합니다.

테스트

1. 각 VM 은 보호를 시작한 후 테스트하고, 이후 최소 월 1 회 테스트합니다.
2. VM 이 부팅되는지, 예상한 주소를 받는지, 애플리케이션이 응답하는지, 게스트 에이전트가 응답하는지 확인합니다.
3. oVirt 가 대상이라면 모든 테스트에 엔진 상태 점검을 포함합니다.

페일오버와 페일백

1. 페일오버는 실제 장애 시에만 실행하고 예약하지 않습니다.
2. 사용자가 대상 네트워크에서 복구된 VM 에 접근할 수 있는지 확인합니다.
3. 페일백은 유지보수 시간대에 계획합니다.

6. 결론

Proxmox VE 와 oVirt 는 같은 KVM 가상 머신을 실행하므로, 두 플랫폼 간의 크로스 하이퍼바이저 DR 은 이미 실용적입니다. 성공의 핵심은 각 플랫폼의 관리 방식을 존중하는 것입니다. Proxmox 는 스스로 복구하고, oVirt 는 엔진을 통해 복구합니다. 스토리지, 네트워크, 엔진 가용성을 미리 계획하고 정기적으로 테스트하면 어느 플랫폼이든 다른 플랫폼을 보호할 수 있습니다.

이 시리즈의 2 부에서는 KVMDR 이 이러한 차이를 어떻게 처리하여, 운영자 입장에서 크로스 하이퍼바이저 복구가 동일한 방식으로 이루어지게 하는지 설명합니다.

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.