KVM/DR

ホワイトペーパー | クロスハイパーバイザー DR シリーズ | 第 2 部（全 2 部）

KVMDR によるクロスハイパーバイザー DR の実現

Proxmox VE、oVirt、OLVM、RHV で統一された復旧体験

KVMDR AI Ltd | kvmdr.ai | 2026 年 11 月

概要

第 1 部では、Proxmox VE と oVirt が同じ KVM 仮想マシンを共有しながらも、管理方式が大きく異なることを示しました。本書では、KVMDR がその差をどのように埋めているかを説明します。プラットフォームに依存しないレプリカ、両端に置くプラットフォーム固有のアダプター、そしてランサムウェアからのポイントインタイム復旧、業務を止めないテスト、柔軟なフェイルバックなど、エンタープライズ DR に求められる復旧機能です。

1. 設計原則：中央は中立に

多くの DR ツールは 1 つのハイパーバイザーを中心に作られ、他は後から追加されています。KVMDR は当初から KVM のために設計されており、保護対象データをどちらのプラットフォームにも属さない形式で保持します。プラットフォーム固有の知識は両端のアダプターだけが持ち、ソースからの読み取り方法とターゲット上での VM 構築方法を担います。



図 1. KVMDR のクロスハイパーバイザー データパス

- **ソースアダプター。** Proxmox（deb）と oVirt（rpm）の両方に向けてパッケージ化された KVMDR ホストエージェントが、各 VM ディスクの変更ブロックを追跡し、リポジトリへ送信します。ゲスト内には何もインストールしません。
- **プラットフォーム中立のリポジトリ。** ディスクは変更履歴のジャーナルを伴うプレーンなイメージとして保持され、CPU、メモリ、ディスク、NIC、ブート設定といった VM の中立的な記述も併せて保持されます。
- **ターゲットアダプター。** Proxmox では、KVMDR は DR ホスト上に直接 VM を構築します。oVirt では、選択したストレージドメインに VM とディスクをエンジン経由で登録します。

KVMDR が各スタックに組み込まれる仕組み

第 1 部で示したとおり、両プラットフォームは QEMU で終わり、QEMU より上はすべて異なります。KVMDR は両者に共通する QEMU 層で変更を捕捉し、各プラットフォームの管理層とは、ターゲット上で VM を構築、起動するときなど必要な場合にのみやり取りします。



図 2. QEMU 層での変更捕捉

1. 初期同期：KVMDR は各ディスクでダーティビットマップを有効化し、その時点以降のすべての書き込みを記録したうえで、読み取り専用の NBD エクスポートを通じてディスク全体を固定サイズのチャンク単位で読み取ります。同期が中断された場合は、最後に完了したチャンクから再開します。
2. 継続的レプリケーション：KVMDR はビットマップに記録されたブロックのみを読み取ってリポジトリのジャーナルに書き込み、新しいビットマップを開始します。フルコピーの開始前にビットマップを有効化しているため、両者の間に書き込みが漏れることはありません。
3. 復旧：ターゲットアダプターが、リポジトリのイメージとジャーナルから指定時点のディスクを生成し、ターゲットプラットフォームに引き渡します。

連携ポイント	Proxmox VE	oVirt、OLVM、RHV
QEMU との通信	Proxmox 自身が使用する QMP ソケット経由で直接	各ホスト上で vdsm および libvirt と併存
VM 設定の読み取り	/etc/pve 内の VM 設定	エンジン REST API
DR ディスクの作成	DR ホストストレージ上のシンボリューム	エンジン経由でターゲットストレージドメイン内にディスクイメージを作成
VM の作成と起動	DR ホスト上の qemu-server	エンジン REST API（vdsm と libvirt に指示）
コンテナ	LVM thin ボリューム上のブロックフィルター	該当なし

> **QEMU 層で捕捉する理由**
> QEMU のダーティビットマップと NBD エクスポートは標準的な QEMU 機能であり、Proxmox Backup Server や oVirt の増分バックアップも同じ機能を利用しています。この層で動作することで、KVMDR は両プラットフォームで同一の捕捉方式を使え、ゲスト内のソフトウェアも不要で、レプリケーション自体はどちらの管理層にも依存しません。

2. クロスハイパーバイザー復旧の流れ

1. 運用担当者が VM または整合性グループと、復旧時点（最新時点、またはランサムウェア発生前のクリーンな時点）を選択します。
2. KVMDR がターゲットの準備状況（ストレージ、ネットワークマッピング、oVirt の場合はエンジン）を確認します。
3. ターゲットアダプターが中立モデルから VM を構築し、そのプラットフォーム向けに名前とネットワークを変換します。
4. 選択した時点のディスクが提示されます。
5. ネットワークルールが適用されます。新しい IP アドレス（Re IP）を割り当てるか、元の MAC アドレスを維持します。

6. 整合性グループ内の VM は、同一時点から同時に起動します。

> **どちらの方向も同じ手順**
> Proxmox から oVirt へ、oVirt から Proxmox へのいずれも、同じワークフローと同じ画面を使用します。第 1 部で説明したプラットフォームの違いは、アダプターが吸収します。

3. 違いを KVMDR が吸収

プラットフォームの違い（第 1 部）	KVMDR の対応
oVirt にはエンジンが必要	エンジン API を通じて VM を作成、起動し、復旧の一環としてエンジンの状態を確認
ディスクはストレージドメイン内に配置	クロスハイパーバイザープロファイルでターゲットのクラスターとストレージドメインを選択
ネットワークモデルの違い	ソースブリッジからターゲットネットワークへのネットワークマップ。Re IP または MAC 維持に対応
命名規則の違い	名前を自動変換（例：Proxmox ではアンダースコアをハイフンに）
コンテナは Proxmox のみ	コンテナは Proxmox から Proxmox へ、VM と同じ機能で保護
テストの仕組みの違い	シミュレーションをプラットフォームごとに最適化：Proxmox では容量効率重視、oVirt ではベースラインを使用

4. DR サイトのサイジング

第 1 部では、oVirt ターゲットでの DR テストには Proxmox より多くの容量が必要だと述べました。その理由は、テスト用に VM の一時的な書き込み可能コピーを作る方法が、プラットフォームごとに異なるためです。

Proxmox で追加容量がほとんど不要な理由

Proxmox ターゲットでは、DR ディスクは KVMDR が直接制御する DR ホスト上のシンボリュームです。シミュレーションでは、KVMDR が瞬時にシンスナップショットを作成し、そこからテスト VM を起動します。新たに容量を消費するのはテスト VM が書き込んだブロックだけで、テスト終了時に破棄されます。その間もレプリケーションは元のディスクに対して継続します。

oVirt でテスト用ベースラインが必要な理由

oVirt ターゲットでは、DR ディスクはストレージドメイン内のイメージであり、スナップショットやクローンを作成できるのはエンジンだけです。第 1 部で示したとおり、oVirt のスナップショットはディスクのボリュームチェーンに qcow2 オーバーレイを追加し、書き込み先をそこへ切り替えます。そのチェーンはエンジンのデータベース上で 1 つの VM に属します。KVMDR がテストのために DR ディスクのスナップショットを作成すると、レプリケーションとテスト VM が同じチェーンを奪い合うことになります。エンジン経由でディスクを独立した新しいイメージにクローンすると、フルコピーになります。

そのため KVMDR は、ストレージドメイン内に独立したテスト用ベースラインを保持します。テスト VM はこのコピーから起動し、その間もレプリケーションは DR ディスクに影響を与えずに継続します。ベースラインのサイズは保護対象ディスクと同じです。

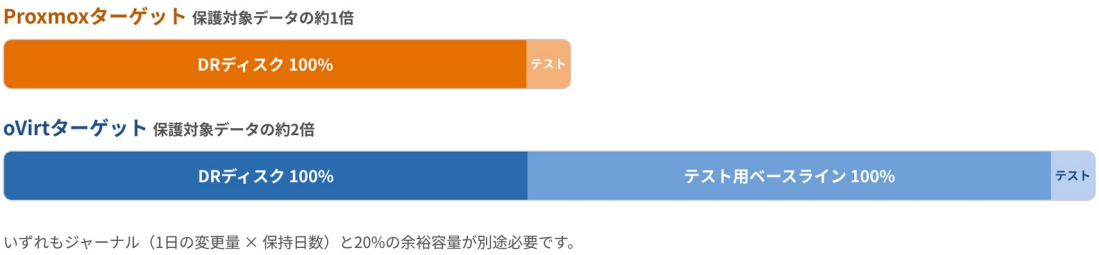


図3. シミュレーション中の DR ターゲット容量

サイジングガイド

項目	Proxmox ターゲット	oVirt ターゲット
DR ディスク	保護対象データの 1 倍	保護対象データの 1 倍
テスト用ベースライン	なし	保護対象データの 1 倍
テスト時の書き込み	少量、一時的	少量、一時的
復旧履歴（ジャーナル）	1 日の変更量 × 保持日数	1 日の変更量 × 保持日数
余裕容量	20%以上	20%以上

> **例：保護対象 VM が 2 TB の場合**
> Proxmox ターゲット：DR ディスク 2 TB＋ジャーナル＋20%の余裕容量。oVirt ターゲット：DR ディスク 2 TB＋テスト用ベースライン 2 TB＋ジャーナル＋20%の余裕容量。oVirt では DR ディスク容量のおよそ 2 倍を計画してください。ジャーナルのサイズは、1 日あたりのデータ変更量と履歴の保持期間によって決まります。

5. プラットフォームを横断するエンタープライズ DR 機能

- **ニア同期レプリケーション。** 変更を継続的に捕捉し、最短 1 秒の鮮度の復旧ポイントを提供します。
- **ポイントインタイム復旧。** 昨夜のバックアップではなく、ランサムウェア被害の直前の時点に戻せます。
- **AI によるランサムウェア検知。** 異常な書き込みパターンを発生時に検出し、最後のクリーンな時点を明示します。
- **シミュレーション。** 保護を継続したまま隔離ネットワーク上で完全な復旧テストを実行でき、本番環境に影響を与えません。
- **整合性グループ。** アプリケーションとそのデータベースなど、関連する VM を同一時点に復旧します。

- **柔軟なフェイルバック。** フェイルオーバー以降の変更分のみを送信して元のサイトに戻すことも、必要に応じて VM やコンテナ全体を復旧することもできます。
- **高可用性の管理。** アクティブとスタンバイの管理サーバーペアにより、サイトを失っても DR の制御が維持されます。
- **ゲスト内エージェント不要。** 保護はハイパーバイザーホストから行います。ゲストに必要なのは、ファイル復元用の標準的な QEMU ゲストエージェントだけです。

6. KVMDR が優れている理由

- **VMware 向け製品の転用ではなく、KVM のために設計。** KVMDR は当初から KVM と QEMU を中心に設計されているため、両プラットフォームを 1 つの中立モデルにきれいに対応付けられます。
- **両プラットフォーム、両方向を 1 つの製品で。** 1 つのライセンスと 1 つのワークフローで、Proxmox VE、oVirt、OLVM、RHV をソースとしてもターゲットとしてもカバーします。
- **コピーではなく復旧を。** ポイントインタイム復旧、Re IP、整合性グループ、シミュレーションにより、レプリカを実際の災害時に使えるものにします。
- **ランサムウェアへの対応。** 検知とクリーンな時点への復旧は、別製品ではなくレプリケーション経路の一部です。
- **エンタープライズ DR の経験。** KVMDR のチームは、エンタープライズ向け災害対策の構築と販売に数十年の経験を持ち、その規律を KVM にもたらしています。

7. 代表的な用途

- **混在環境。** メインのデータセンターでは oVirt または OLVM、セカンダリサイトでは Proxmox を使い、互いを保護します。
- **プラットフォームの移行。** oVirt や RHV から Proxmox へ移行するチームは、レプリケーション、シミュレーションによるテストを経て切り替えができ、ソースはロールバック用に残せます。
- **低コストの DR サイト。** oVirt または OLVM の本番環境を、汎用ハードウェア上の Proxmox DR サイトで保護します。

8. まとめ

Proxmox VE と oVirt の管理方式は大きく異なりますが、動かしている仮想マシンは同じです。KVMDR はその共通基盤を活かし、保護対象データを中立に保ち、プラットフォーム固有の知識を両端のアダプターに置き、それをエンタープライズ DR 機能で包んでいます。その結果、すべての KVM プラットフォームで、どちらの方向にも統一された復旧体験を提供します。

実際に動作をご確認ください。kvmdr.ai で 7 日間の無料トライアルを開始するか、デモをご予約いただけます。

KVMDR について

KVMDR AI Ltd は、KVM ベースのハイパーバイザー（Proxmox VE、oVirt、Oracle Linux Virtualization Manager、Red Hat Virtualization）向けの災害対策（DR）製品を開発する英国企業です。KVMDR は、ニア同期レプリケーション、ランサムウェアからのポイントインタイム復旧、業務を止めない DR テスト、AI による脅威検知を 1 つのプラットフォームで提供します。7 日間の無料トライアルは kvmdr.ai から開始できます。