뉴시스 (NEWsys)

Proxmox → Unraid 마이그레이션 상세 조사 보고서

Proxmox → Unraid 가상머신 마이그레이션 상세 조사 보고서

문서번호 NEWsys-OPS-2026-1103-01 **작성일** 2026-11-03 **대상** Proxmox `newsyst` (192.168.0.9) → Unraid `NS` (192.168.0.113) **작성** 뉴시스 (NEWsys) · 대표 박재우 · 사업자등록번호 513-18-56586

---

1. 조사 개요

Proxmox VE에서 운영 중인 가상머신 7대를 Unraid로 이전할 수 있는지, 이전한다면 어떤 방법과 절차가 필요한지 **양쪽 서버에 직접 접속해 실측**한 결과를 정리했다.

결론을 먼저 적으면 다음과 같다.

> **기술적으로 전부 이전 가능하다.** 다만 하이퍼바이저 구조가 달라 **디스크 형식 변환 · 디스크 버스 변경 · 네트워크 브리지 변경** 3가지를 반드시 거쳐야 하고, **스냅샷 9개는 이전되지 않는다.** 전 VM 이전 시 예상 다운타임은 **약 3~4시간**이다.

---

2. 실측 인벤토리

2.1 Proxmox VE (이전 원본)

항목	값
호스트명	`newsyst` (192.168.0.9)
버전	**PVE 9.2.24** (kernel 7.0.14-26-pve)
CPU	80 스레드
메모리	125GB (사용 30GB)
스토리지	`local` (dir, 94GB) · **`local-lvm` (lvmthin, 6.4TB)**
VM 수	**7대 (전부 running)**

2.2 가상머신 상세

VMID	이름	OS	vCPU	RAM	디스크	실제 사용	NIC
100	NEWsys-Agent	Linux	8	8GB	100GB	17.8GB	**e1000e**

VMID	이름	OS	vCPU	RAM	디스크	실제 사용	NIC
101	NEWsys-DEV-Server	Linux	8	8GB	100GB	25.0GB	**e1000e**
102	Rocky-10-2	Linux	8	8GB	100GB	8.3GB	**e1000e**
103	Rocky-work	Linux	8	8GB	100GB	6.8GB	**e1000e**
104	Win11IoTEntLTSC64Kor	Windows 11	8	8GB	100GB + EFI/TPM	33.9GB	virtio
105	OA-jwpark	Windows 11	8	8GB	100GB + EFI/TPM	37.6GB	virtio
106	DEV	Linux	8	8GB	100GB + cloudinit	12.1GB	virtio

합계 — 할당 700GB, **실제 사용 141.5GB** (thin provisioning) **vCPU 총합** 56 / 호스트 80 · **RAM 총합** 56GB / 호스트 125GB

2.3 디스크 구성 (전 VM 공통)

```
scsi0: local-lvm:vm-XXX-disk-0, iothread=1, size=100G, ssd=1
```

- 버스: **SCSI** (virtio-scsi)
- 저장: **LVM-thin 논리볼륨** (파일이 아니라 블록 디바이스 `/dev/pve/vm-XXX-disk-0`)
- Windows 2대는 추가로 **`efidisk0` (4MB) + `tpmstate0` (4MB)** 사용
- Windows 2대는 **OVMF(UEFI) + machine `pc-q35-11.0`**, CPU 모델 `x86-64-v4`
- Linux 1대(106)는 cloud-init 볼륨 사용

2.4 스냅샷 현황

```
snap_vm-100-disk-0_preupgrade-20261029
snap_vm-102-disk-0_preupgrade-20261029
snap_vm-103-disk-0_preupgrade-20261029
snap_vm-104-disk-1_preupgrade-20261029
snap_vm-105-disk-1_preupgrade-20261029   (총 9개)
```

2026-10-29 업그레이드 전 스냅샷이다. **LVM-thin 스냅샷은 Unraid로 이전되지 않는다.** 이전 전에 반드시 병합(merge)하거나 삭제해야 한다.

2.5 Unraid (이전 대상)

항목	값
호스트명	`NS` (192.168.0.113)
버전	**Unraid 7.3.3** (kernel 6.18.54-Unraid)
CPU	128 스레드
메모리	251GB (사용 16GB)

항목	값
배열	22TB (사용 6.4TB, **여유 16TB**) — HDD 3.6TB ×6
캐시	NVMe 931GB (사용 341GB, **여유 591GB**)
libvirt	**12.2.0 실행 중**
브리지	`br0` 활성 (물리 MAC 9c:6b:00:5a:08:71)
현재 VM	3대 (전부 shut off: PacketFence, Windows 11, Windows 11_2)
VM 저장소	`/mnt/user/domains` (비어 있음)

용량 판단 — 실제 이전 대상 **141.5GB**는 **캐시 NVMe 여유 591GB에 충분히 들어간다.**

---

3. 구조 차이 분석

이전이 "복사"로 끝나지 않는 이유는 두 하이퍼바이저의 **저장 방식이 다르기** 때문이다.

구분	Proxmox VE	Unraid
하이퍼바이저	KVM + 자체 관리(pvesm)	KVM + **libvirt**
디스크 저장	**LVM-thin 논리볼륨** (블록)	**파일** (`.img` / `.qcow2`)
디스크 경로	`/dev/pve/vm-100-disk-0`	`/mnt/user/domains/<VM>/vdisk1.img`
기본 디스크 버스	**SCSI (virtio-scsi)**	**virtio-blk** 또는 SATA
네트워크 브리지	`vmbr0`	`br0`
스냅샷	LVM-thin / ZFS 스냅샷	qcow2 내부 스냅샷(제한적)
UEFI 지원	OVMF	OVMF (지원)
VM 관리	`qm` CLI / Web UI	libvirt XML / Web UI

3.1 반드시 변환해야 하는 3가지

① 블록 디바이스 → 파일

LVM 논리볼륨은 파일이 아니다. Unraid가 읽을 수 있는 파일로 **변환**해야 한다.

```
# LVM LV → qcow2 (스냅샷 지원, 용량 유연)
qemu-img convert -p -O qcow2 /dev/pve/vm-100-disk-0 /tmp/vm-100.qcow2

# 또는 raw (더 빠르지만 용량 고정 100GB)
qemu-img convert -p -O raw /dev/pve/vm-100-disk-0 /tmp/vm-100.raw
```

> **raw를 권장**한다. Unraid는 raw vdisk에서 성능이 가장 좋고, qcow2는 스냅샷을 쓰지 않으면 이점이 적다. 다만 raw는 실제 사용량과 무관하게 100GB를 차지한다 — 7대면 700GB로 캐시(591GB)를 넘는다. **캐시에 넣을 거면 qcow2(실제 144GB), 배열에 넣을 거면 raw**가 현실적이다.

② 디스크 버스 변경

Proxmox의 `scsi`(virtio-scsi)는 Unraid에서 기본 제공되지 않는다. Unraid VM 생성 시 **virtio-blk** 또는 **SATA**를 선택해야 한다.

- **Linux**: virtio-blk 권장. 커널에 `virtio_blk` 모듈이 있으면 그대로 부팅된다 (대부분 포함).
- **Windows**: virtio-blk로 바꾸면 **virtio-win 드라이버가 없을 경우 부팅 불가 (INACCESSIBLE_BOOT_DEVICE)**. 안전하게 **SATA**로 시작하고, 부팅 후 virtio 드라이버를 설치한 다음 virtio-blk로 전환한다.

③ 네트워크 브리지 변경

`vmbr0` → `br0`. MAC 주소는 그대로 유지하면 게스트 OS의 네트워크 설정(고정 IP)이 그대로 동작한다.

3.2 CPU 모델 주의

Proxmox Windows 2대는 CPU 모델이 **`x86-64-v4`** 로 고정되어 있다. Unraid 기본값은 `host-passthrough` 이므로 그대로 두면 문제없다. 다만 XML에 `x86-64-v4`를 그대로 복사하면 Unraid 호스트 CPU가 v4를 지원하지 않을 경우 부팅이 실패한다.

```
<!-- Unraid에서는 이렇게 (권장) -->
<cpu mode='host-passthrough' check='none' migratable='off'/>
```

3.3 NIC 모델 주의

Linux VM 4대는 **e1000e**를 쓰고 있다. Unraid는 e1000을 지원하지만 성능이 낮다.

방법	난이도	비고
e1000 유지	쉬움	그대로 부팅, 성능 손해
virtio로 전환	중간	Linux는 대부분 자동 인식 (`virtio_net`)

Linux 4대는 virtio_net으로 바꿔도 무리 없이 잡힌다. 다만 **고정 IP를 인터페이스 이름(ens18 등)으로 설정**했다면 이름이 바뀌어 네트워크가 끊길 수 있다. 사전에 확인이 필요하다.

---

4. 이전 방법 비교

방법 A — qemu-img 변환 + 파일 복사 (권장)

```
[Proxmox]                              [Unraid]
VM 정지
  ↓
스냅샷 병합
  ↓
qemu-img convert
  ↓
qcow2/raw 파일 생성
  ↓  scp / rsync (1Gbps)
/mnt/user/domains/<VM>/vdisk1.img
  ↓
Unraid VM 생성 (XML)
  ↓
부팅 · 네트워크 확인
```

항목	내용
장점	가장 확실, VM 단위 통제 가능
단점	VM별 수작업, 변환 시간 소요
다운타임	VM당 **20~40분** (100GB 중 15~40GB 실제)
필요 공간	Proxmox 임시 144GB + Unraid 144GB

방법 B — vzdump 백업 → 복원

항목	내용
장점	Proxmox 표준 백업 사용
단점	`.vma` **형식은 libvirt가 직접 못 읽음** → 결국 `vma extract` 후 다시 변환
판단	**권장하지 않음.** 단계만 늘고 이득이 없다

방법 C — 라이브 마이그레이션

Proxmox ↔ Unraid 간 라이브 마이그레이션은 **불가능**하다. 공유 스토리지와 동일 하이퍼바이저가 전제다.

> **정리: 방법 A로 진행한다.**

---

5. 상세 절차

5.1 사전 준비 (다운타임 없음)

```
# ① Unraid에 도메인 공유 생성 (VM 디스크용)
#    Unraid WebUI → Shares → Add Share
#    이름: domains
#    Primary storage: cache (NVMe)  ← 속도
#    Secondary storage: array       ← 용량
#    Allocation: high-water

# ② Proxmox 임시 공간 확보 확인
ssh root@192.168.0.9 'df -h /tmp; pvesm status | grep local-lvm'

# ③ Unraid 수신 공간 확인
ssh root@192.168.0.113 'df -h /mnt/cache /mnt/user'

# ④ 전 VM 백업 (안전장치)
#    Proxmox에 별도 디스크가 없으므로 Unraid 배열로 직접 저장
ssh root@192.168.0.9 'vzdump 100 --dumpdir /mnt/pve/unraid-backup --mode stop'  # NFS 필요
```

5.2 VM별 이전 (VM당 20~40분)

예시 — VM 100 (Linux)

```
# ① 스냅샷 병합 (이전 안 됨)
ssh root@192.168.0.9 'qm listsnapshot 100'
ssh root@192.168.0.9 'qm delsnapshot 100 preupgrade-20261029'

# ② VM 정지
ssh root@192.168.0.9 'qm shutdown 100 && sleep 10 && qm status 100'

# ③ 파일로 변환 (raw 100GB)
ssh root@192.168.0.9 'qemu-img convert -p -O raw \
  /dev/pve/vm-100-disk-0 /tmp/vm-100.raw'

# ④ 압축 후 전송 (100GB → 실제 18GB 수준)
ssh root@192.168.0.9 'gzip -1 -c /tmp/vm-100.raw > /tmp/vm-100.raw.gz'
rsync -av --progress root@192.168.0.9:/tmp/vm-100.raw.gz /mnt/user/domains/NEWsys-Agent/

# ⑤ 수신 측 해제
ssh root@192.168.0.113 'cd /mnt/user/domains/NEWsys-Agent && \
  gunzip vm-100.raw.gz && mv vm-100.raw vdisk1.img'

# ⑥ Proxmox 임시 파일 삭제
ssh root@192.168.0.9 'rm -f /tmp/vm-100.raw /tmp/vm-100.raw.gz'
```

Unraid에서 VM 생성 — WebUI `VMS → Add VM` 또는 XML 직접 작성

```
<domain type='kvm'>
  <name>NEWsys-Agent</name>
  <memory unit='KiB'>8388608</memory>
  <vcpu>8</vcpu>
  <os>
    <type arch='x86_64' machine='pc-q35-9.2'>hvm</type>
  </os>
  <devices>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source file='/mnt/user/domains/NEWsys-Agent/vdisk1.img'/>
      <target dev='vda' bus='virtio'/>
    </disk>
    <interface type='bridge'>
      <source bridge='br0'/>
      <mac address='BC:24:11:8D:F9:2D'/>   <!-- 원본 MAC 유지 -->
      <model type='virtio'/>
    </interface>
  </devices>
</domain>
```

Windows VM 추가 처리 (104, 105)

```
 <!-- UEFI 필수 -->
<os>
  <type arch='x86_64' machine='pc-q35-9.2'>hvm</type>
  <loader readonly='yes' type='pflash'>/usr/share/qemu/ovmf-x64/OVMF_CODE-pure-efi.fd</
loader>
  <nvram>/etc/libvirt/qemu/nvram/OA-jwpark_VARS-pure-efi.fd</nvram>
</os>
<!-- Win11은 TPM 2.0 필수 (Proxmox의 tpmstate0에 해당) -->
<tpm model='tpm-crb'>
  <backend type='emulator' version='2.0'/>
</tpm>

<!-- 디스크는 처음엔 SATA로 (virtio 드라이버 미설치 상태 대비) -->
<disk type='file' device='disk'>
  <driver name='qemu' type='raw'/>
  <source file='/mnt/user/domains/OA-jwpark/vdisk1.img'/>
  <target dev='sda' bus='sata'/>
</disk>
```

> Windows는 Proxmox에서 쓰던 **efidisk0(4MB)와 tpmstate0(4MB)** 도 함께 옮겨야 한다. efidisk가 없으면 `Boot Failed: not a bootable disk` 가 뜬다. TPM은 Proxmox의 상태를 그대로 옮길 수 없으므로 **Unraid에서 새로 생성(emulator)** 한다. Win11은 TPM 2.0이 없으면 부팅은 되지만 일부 보안 기능과 업데이트에서 문제가 생길 수 있다.
>
> Unraid 펌웨어 경로 (실측 확인): `/usr/share/qemu/ovmf-x64/OVMF_CODE-pure-efi.fd ← TPM 미사용 /usr/share/qemu/ovmf-x64/OVMF_CODE-pure-efi-tpm.fd ← TPM 사용 (Win11)`

5.3 이전 후 확인

```
# VM 상태
ssh root@192.168.0.113 'virsh list --all'

# 게스트 응답 확인 (Linux)
ping -c 3 <VM_IP>

# 디스크 인식 확인 (게스트 내부)
lsblk
df -h
```

---

6. 소요 시간 추정

단계	VM당	7대 합계
스냅샷 병합	1~2분	10분
VM 정지	1분	7분
qemu-img 변환	5~15분	70분
압축 + 전송 (1Gbps)	5~10분	50분
Unraid VM 생성	5분	35분
부팅·검증	5분	35분
합계	**약 30분**	**약 3.5시간**

> 전송은 1Gbps 기준 실효 100MB/s. gzip 압축 시 141.5GB → 약 70~90GB로 줄어 실제 전송은 50분 내외다.

---

7. 리스크와 대응

리스크	발생 가능성	영향	대응
Windows 부팅 실패 (virtio 드라이버)	**높음**	중	SATA 버스로 시작 → 드라이버 설치 후 전환
efidisk 누락으로 부팅 불가	**높음**	중	efidisk·tpmstate 함께 변환
Linux 고정 IP 끊김	중	중	MAC 유지 + 사전 인터페이스 이름 확인
캐시 NVMe 용량 초과	중	중	qcow2 사용 또는 배열로 분산
스냅샷 유실	**확실**	낮음	이전 전 병합, 필요 시 별도 백업
원본 손상	낮음	**높음**	Proxmox VM은 삭제하지 않고 보존 (롤백용)
캐시 디스크 단일 장애	중	**높음**	도메인 공유를 cache+array 이중화 또는 정기 백업

---

8. 롤백

원본을 지우지 않으면 롤백은 즉시 가능하다.

```
# Unraid에서 문제 발생 시
ssh root@192.168.0.113 'virsh destroy NEWsys-Agent'
# → Proxmox에서 재기동
ssh root@192.168.0.9 'qm start 100'
```

원칙: 7대 전부 이전이 끝나고 2주간 안정 운영이 확인될 때까지 Proxmox VM을 삭제하지 않는다.

---

9. 판단 — 전부 옮겨야 하는가

이전은 가능하지만, **무조건 옮기는 것이 답은 아니다.** 두 서버의 성격이 다르다.

구분	Proxmox	Unraid
VM 성능	**우수** (LVM-thin, virtio-scsi, iothread)	양호 (파일 기반 vdisk)
스냅샷	**강력** (LVM/ZFS)	제한적
백업	vzdump / PBS	수동·스크립트
스토리지 확장	디스크 추가 후 LVM	**매우 쉬움** (배열)
컨테이너	LXC	**Docker 특화**
관리 편의	중	**쉬움**

권고안

VM	이전 판단	이유
100 NEWsys-Agent	**보류**	에이전트 상시 구동 — 성능·안정성 우선
101 NEWsys-DEV-Server	이전 가능	개발용, 다운타임 허용
102 Rocky-10-2	이전 가능	테스트 VM
103 Rocky-work	이전 가능	테스트 VM
104 Win11 IoT	이전 가능	단, UEFI·드라이버 검증 필수
105 OA-jwpark	**보류**	업무용 — 실패 시 업무 중단
106 DEV	이전 가능	개발용

1단계(권장) — 다운타임이 허용되는 VM 3대(101, 102, 103)만 먼저 옮겨 절차를 검증한다. **2단계** — Windows 1대(104)로 UEFI·드라이버 경로를 검증한다. **3단계** — 나머지 판단.

> 한 번에 7대를 옮기면 문제 발생 시 원인 분리가 어렵다. **소수부터 검증**하는 것이 정석이다.

---

10. 결론

1. **기술적 이전은 가능하다.** 장애물은 3가지 변환(블록→파일, SCSI→virtio/SATA, vmbr0→br0)뿐이다.
2. **실제 데이터는 141.5GB**로 예상보다 작다. 캐시 NVMe(591GB 여유)에 전부 수용 가능하다.
3. **스냅샷 9개는 이전되지 않는다.** 사전 병합이 필수다.
4. **Windows 2대가 최대 변수**다. UEFI·efidisk·virtio 드라이버를 반드시 검증해야 한다.
5. **전부 옮기는 것은 권장하지 않는다.** 테스트 VM 3대로 절차를 검증한 뒤 확대하는 것이 안전하다.
6. **원본은 2주간 보존**한다. 롤백은 Proxmox VM을 켜면 끝난다.

실증 결과 (2026-11-03)

테스트 VM 1대(8.2GB)를 실제로 이전해 **약 8분**에 완료했다. 부팅은 성공했으나 **NIC 이름 고정(`interface-name=ens18`)** 때문에 네트워크가 붙지 않았고, 오프라인 마운트로 한 줄을 제거해 해결했다. MAC을 유지했기 때문에 IP까지 그대로 복구되었다.

> 절차는 검증되었다. 나머지 VM은 **`interface-name=` 사전 제거**만 추가하면 동일하게 진행할 수 있다.

---

11. 실증 파일럿 결과 (2026-11-03 실행)

보고서 작성 후 **실제로 1대를 이전**해 절차를 검증했다. 대상은 다운타임이 허용되는 테스트 VM(102 Rocky-10-2)으로 잡았다. 원본은 삭제하지 않아 즉시 롤백이 가능한 상태로 진행했다.

11.1 실행 조건

항목	값
대상 VM	102 Rocky-10-2 (Rocky Linux, hostname `NEWsys-DEV`)
할당 디스크	100GB (thin)
실제 데이터	**8.22GB**
vCPU / RAM	8 / 8GB
NIC	e1000e · MAC `BC:24:11:97:9A:26`
원본 IP	192.168.0.148
Proxmox	192.168.0.9 → Unraid 192.168.0.113 (1Gbps)

11.2 실제 소요 시간 (측정값)

단계	시간	비고
VM 정지	8초	`qm shutdown`
qcow2 변환	약 40초	`qemu-img convert -O qcow2` (8.3GB)
파일 전송	**1분 18초**	rsync · **112MB/s**
raw 변환 (Unraid)	9초	`qemu-img convert -O raw`
VM 정의·기동	10초	`virsh define` + `start`
네트워크 복구	**약 5분**	아래 11.3 문제 해결 포함
합계	**약 8분**	VM 1대 · 8.2GB 기준

> 100GB 할당이지만 실제 데이터가 8.2GB였기 때문에 8분에 끝났다. 실제 데이터량이 시간을 결정한다.

11.3 발생한 문제 — NIC 이름 고정

증상 — VM은 정상 부팅해 로그인 화면까지 올라왔으나 **네트워크가 응답하지 않았다.** `ping` 손실 100%, ARP 조회 실패.

원인 — 게스트의 NetworkManager 프로필이 Proxmox 시절 NIC 이름에 고정되어 있었다.

```
# /etc/NetworkManager/system-connections/ens18.nmconnection
[connection]
id=ens18
interface-name=ens18      ← 이 줄이 문제
```

Unraid에서는 NIC 이름이 달라(`ens3` / `enp1s0` 등) 이 프로필이 어떤 장치에도 매칭되지 않아 DHCP 요청 자체가 나가지 않았다.

해결 — VM을 정지하고 디스크를 **오프라인으로 마운트**해 설정을 수정했다. 게스트 콘솔 로그인 없이 처리할 수 있어 편리하다.

```
# Unraid에서
modprobe nbd max_part=8
qemu-nbd -c /dev/nbd0 -f raw /mnt/user/domains/Rocky-10-2/vdisk1.img
vgchange -ay rl
mount /dev/rl/root /mnt/guest

F=/mnt/guest/etc/NetworkManager/system-connections/ens18.nmconnection
grep -v '^interface-name=' "$F" > "$F.tmp" && mv "$F.tmp" "$F"
chmod 600 "$F"

umount /mnt/guest && vgchange -an rl && qemu-nbd -d /dev/nbd0
```

`interface-name=` 한 줄을 제거하면 MAC 주소 기준으로 어떤 이더넷 장치에도 매칭된다. 재기동 후 **원래 IP(192.168.0.148)를 그대로 되찾았다.**

> **사전에 하면 다운타임을 줄일 수 있다.** VM을 끄기 전에 원본 디스크를 마운트해 `interface-name=` 을 미리 제거해 두면, 부팅 후 바로 네트워크가 붙는다.

11.4 최종 검증

항목	결과
VM 상태 (Unraid)	`running`
ping 192.168.0.148	**0% 손실** (3/3)
SSH (22)	OPEN
MAC 주소	원본 유지 (`BC:24:11:97:9A:26`)
IP 주소	원본 유지 (192.168.0.148)
디스크 사용 (Unraid)	8.2GB (100GB sparse)
원본 (Proxmox)	LV·스냅샷 보존 — `qm start 102` 로 즉시 롤백

11.5 검증에서 얻은 교훈

1. **최대 함정은 디스크가 아니라 네트워크였다.** SCSI(virtio-scsi) 버스는 그대로 두어도 Rocky Linux가 문제없이 부팅했다. 반면 NIC 이름 고정은 부팅 성공 후에도 서비스를 죽인다.
2. **MAC 유지가 IP 보존의 핵심이다.** MAC을 바꿨다면 고정 IP를 쓰는 VM은 전부 재설정이 필요하다.
3. **오프라인 마운트가 가장 빠른 복구 수단이다.** 게스트 계정을 모르거나 콘솔 조작이 어려울 때 `qemu-nbd` + LVM 마운트로 디스크를 직접 고칠 수 있다.
4. **thin 프로비저닝 덕에 시간이 크게 줄었다.** 100GB 할당이 8.2GB 전송으로 끝났다.

11.6 남은 VM 적용 시 보정 사항

이번 결과를 반영해 나머지 VM 절차를 다음과 같이 보정한다.

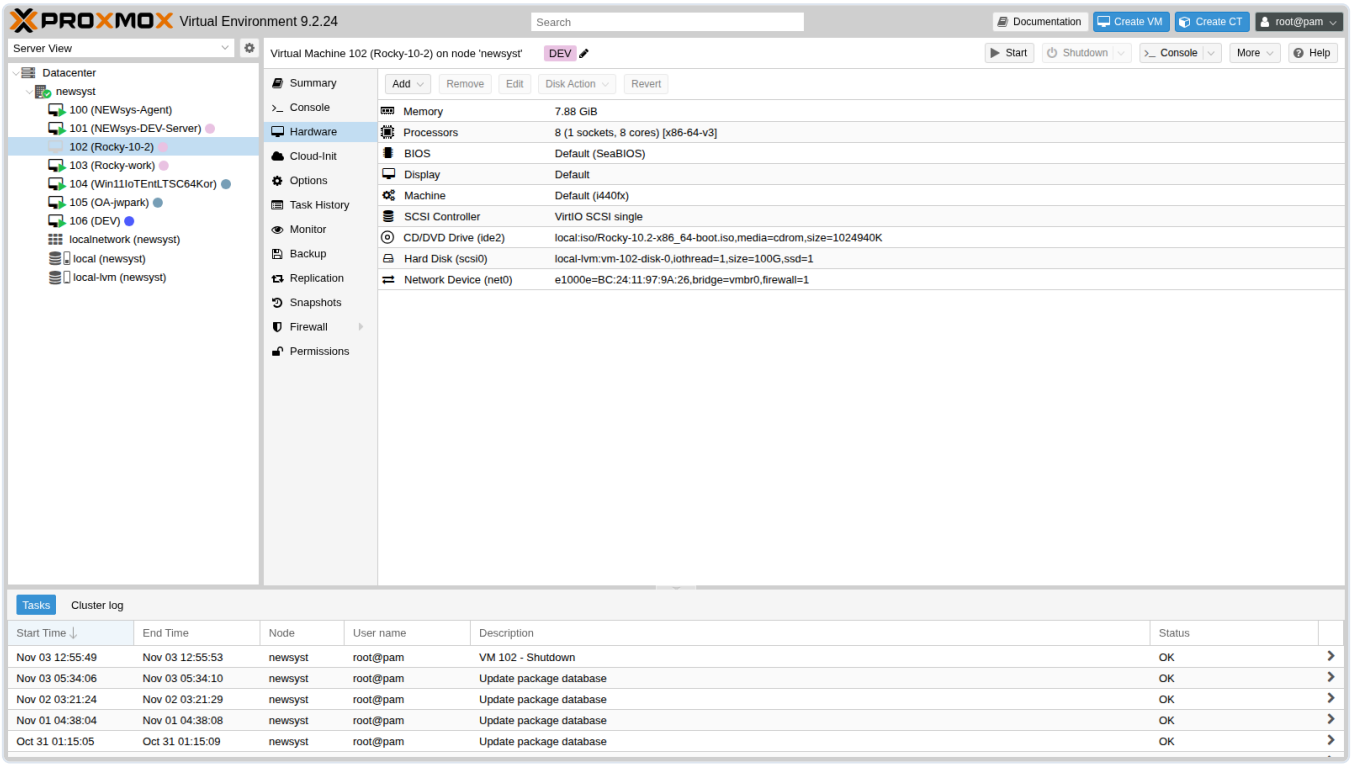
단계	보정 내용
이전 전	**`interface-name=` 사전 제거** (다운타임 단축)
Windows 2대	efidisk·TPM 별도 처리 + virtio 드라이버 사전 주입 검토
검증	ping → SSH → 서비스 포트 순으로 확인
롤백	양쪽 VM 동시 기동 금지 (IP 충돌)

---

12. 실행 화면

실제 작업한 화면을 그대로 첨부한다. (2026-11-03 파일럿)

12.1 이전 전 — Proxmox VM 102 하드웨어 구성



메모리 7.88GiB · vCPU 8 · SCSI 컨트롤러 VirtIO SCSI single · 하드디스크 `local-lvm:vm-102-disk-0` (100G) · 네트워크 `e1000e`, bridge `vmbr0`. 이 세 가지(버스·디스크 형식·브리지)가 변환 대상이다.

12.2 파일 변환 — LVM 논리볼륨을 파일로

```
root@newsyst (Proxmox 192.168.0.9)

$ qm shutdown 102
status: stopped

$ qemu-img info /dev/pve/vm-102-disk-0 | head -3
image: /dev/pve/vm-102-disk-0
file format: raw
virtual size: 100 GiB (107374182400 bytes)

$ qemu-img convert -O qcow2 /dev/pve/vm-102-disk-0 /var/tmp/vm-102.qcow2
(진행 표시줄 100% 완료)

$ ls -lh /var/tmp/vm-102.qcow2
-rw-r--r-- 1 root root 8.3G Nov  3 12:56 /var/tmp/vm-102.qcow2

$ qemu-img info /var/tmp/vm-102.qcow2 | head -4
image: /var/tmp/vm-102.qcow2
file format: qcow2
virtual size: 100 GiB
disk size: 8.22 GiB
```

VM 정지 후 `qemu-img convert`로 qcow2 변환. 할당 100GB지만 thin 프로비저닝으로 실제는 8.22GB였다.

12.3 전송 — 112MB/s

```
Proxmox 192.168.0.9  →  Unraid 192.168.0.113

$ rsync -a --info=progress2 /var/tmp/vm-102.qcow2 \
    root@192.168.0.113:/mnt/user/domains/Rocky-10-2/

  8,822,980,608 100%  112.11MB/s  0:01:15 (xfr#1, to-chk=0/1)

real  1m17.856s

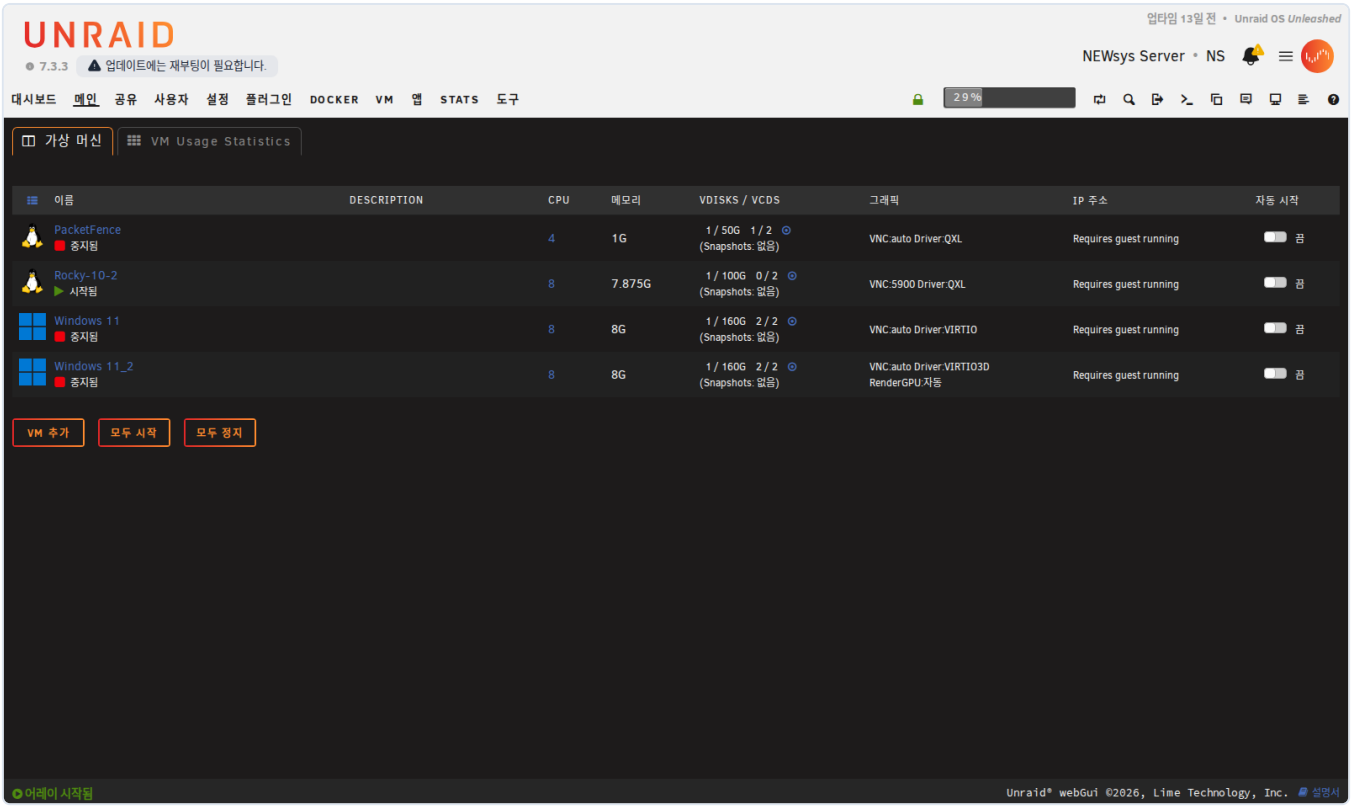
--- 수신 확인 (Unraid) ---
$ ls -lh /mnt/user/domains/Rocky-10-2/
total 8.3G
-rw-r--r-- 1 root root 8.3G Nov  3 12:56 vm-102.qcow2

--- raw 변환 ---
$ qemu-img convert -O raw vm-102.qcow2 vdisk1.img
real  0m9.279s

$ qemu-img info vdisk1.img | head -4
image: vdisk1.img
file format: raw
virtual size: 100 GiB
disk size: 8.19 GiB   (sparse)
```

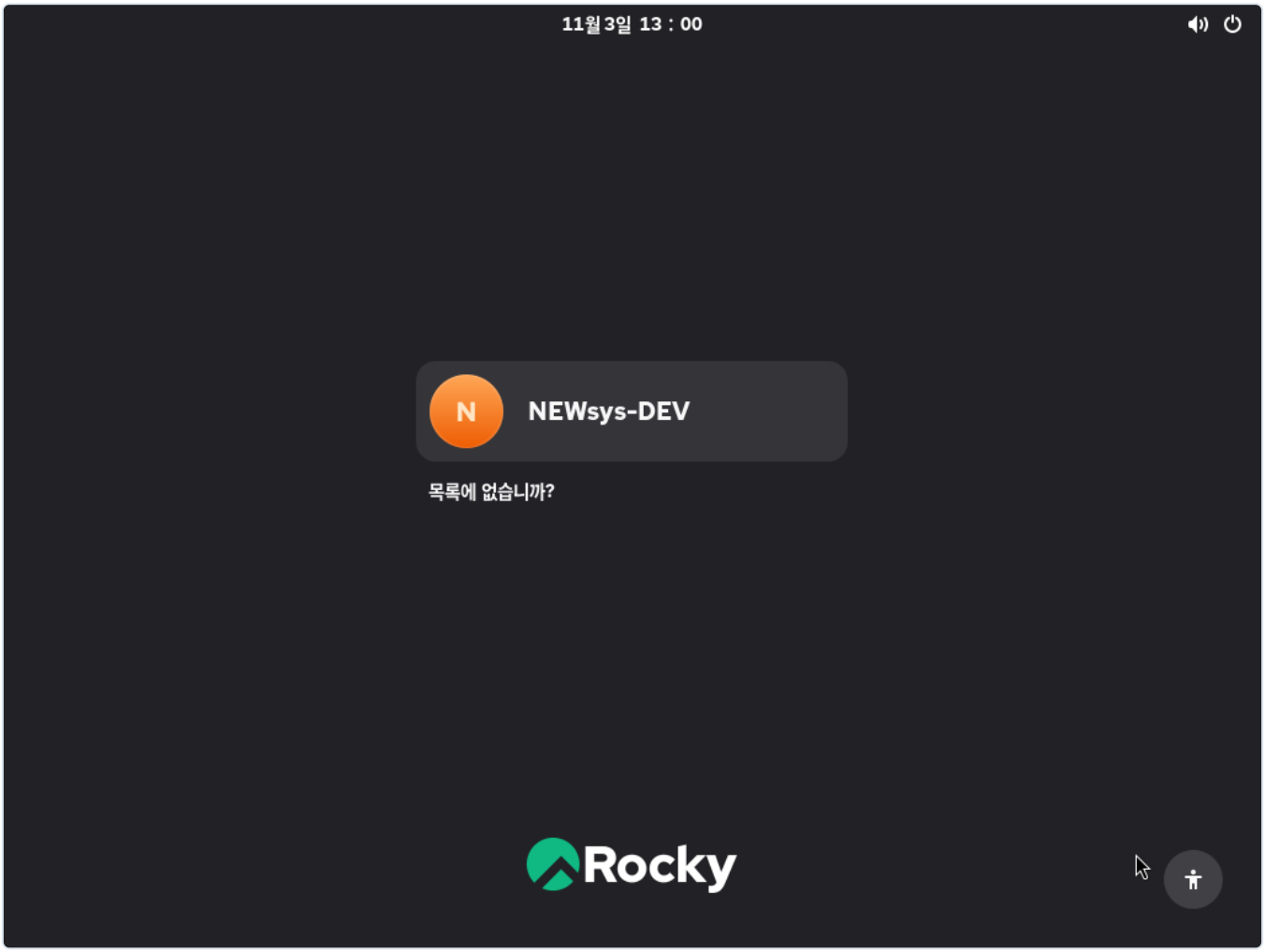
8.8GB를 1분 18초에 전송. Unraid에서 raw로 다시 변환(9초)해 `vdisk1.img` (sparse 100GB) 생성.

12.4 이전 후 — Unraid에서 기동



`Rocky-10-2`가 Unraid에서 시작됨 상태. vDisk 100G 중 8G 할당, VNC 5900으로 콘솔 접근.

12.5 부팅 성공 — 그러나 네트워크 없음



게스트 OS는 정상 부팅해 로그인 화면까지 올라왔다. 문제는 그다음이었다 — ping 100% 손실.

12.6 원인과 해결 — NIC 이름 고정

```
부팅은 성공 · 네트워크 응답 없음 → 원인 추적

$ ping -c 3 192.168.0.148
3 packets transmitted, 0 received, 100% packet loss

$ cat /mnt/guest/etc/NetworkManager/system-connections/ens18.nmconnection
[connection]
id=ens18
interface-name=ens18        ← Proxmox 시절 NIC 이름에 고정
type=ethernet
[ipv4]
method=auto

--- Unraid에서는 NIC 이름이 달라 매칭되지 않음 (ens3 / enp1s0) ---

$ grep -v '^interface-name=' ens18.nmconnection > tmp && mv tmp ens18.nmconnection
$ chmod 600 ens18.nmconnection

--- 재기동 후 ---
2 packets transmitted, 2 received, 0% packet loss
SSH(22) OPEN
IP 192.168.0.148 복구 · MAC BC:24:11:97:9A:26 유지
```

interface-name=ens18 한 줄이 원인이었다. 오프라인 마운트로 제거한 뒤 재기동하자 원래 IP로 복구되었다.

부록 A. 사전 점검 명령 모음

```
# Proxmox — 스냅샷 목록
for id in 100 101 102 103 104 105 106; do echo "VM $id:"; qm listsnapshot $id; done

# Proxmox — 실제 사용량
lvs --noheadings -o lv_name,lv_size,data_percent | grep vm-

# Proxmox — VM 설정 백업 (XML)
for id in 100 101 102 103 104 105 106; do qm config $id > /root/vm-$id.conf; done

# Unraid — 수용 가능 공간
df -h /mnt/cache /mnt/user
virsh list --all
ip -br link show type bridge
```

부록 B. Linux 게스트 사전 확인

이전 전에 게스트 안에서 아래를 확인하면 부팅 실패를 예방할 수 있다.

```
# virtio_blk 모듈 포함 여부 (initramfs)
lsinitramfs /boot/initrd.img-$(uname -r) | grep virtio_blk

# 네트워크 설정 방식 확인 (인터페이스 이름 의존 여부)
cat /etc/netplan/*.yaml 2>/dev/null || cat /etc/network/interfaces 2>/dev/null

# 디스크 식별 방식 (UUID 기반이면 안전)
cat /etc/fstab
```

> `fstab`이 `/dev/sda1` 같은 **디바이스 이름**으로 되어 있으면 이전 후 부팅이 실패한다. `UUID=` 또는 `LABEL=` 기반이어야 안전하다.

문의 뉴시스 (NEWsys) · 053-327-1917 · jwpark@newsyst.com