01 / PREPARE - VIDEO 0:00-4:43

Old drives. Useful backups.

Build a two-drive ZFS mirror and use it as local Proxmox backup storage. Tech without the technobabble.

The result

Two drives hold the same data. A ZFS dataset provides a folder for backup archives; Proxmox registers it as Directory storage. Two 1 TB drives provide roughly one drive's capacity, with less available after overhead.

1. Get ready

- ☐ A working Proxmox VE host, its web interface and root access to the host Shell.
- ☐ Two spare SATA drives, data cables, power connectors and secure mounting hardware.
- ☐ A separate copy of anything valuable on the selected drives.
- ☐ A small test VM and enough separate VM disk storage for a restore test.

Shut down the host, disconnect its power and fit the drives securely with airflow. Connect SATA power and data, then boot. A loose drive is not an advanced vibration-management system.

A mirror is hardware protection

Writes go to both members; reads can be served from either. Deletion affects both. Backups stored here help restore guests, but this host and its mirror share risks such as theft, fire and power damage. Keep another backup copy on a separate device or off-site.

Read through first. Commands run on the **Proxmox host**, not inside a VM. Run each block separately and check the result before continuing. Names in CAPITALS are placeholders to replace.

02 / IDENTIFY - VIDEO 5:00-5:59 AND 11:17-12:49

Choose the right two drives

Device letters can change. Serial numbers and stable device IDs are your friends.

2. Inventory the host

```
pveversion -v
lsblk -o NAME,SIZE,MODEL,SERIAL,FSTYPE,MOUNTPOINTS
findmnt /
pvesm status
pvs
lvs
zpool status
```

Match each drive by model, size and serial to its physical label. Check the tree beneath each disk for partitions and mounts. pvs/lvs reveal LVM use; zpool status shows active ZFS pools. An unmounted disk can still belong to a live storage system.

3. Save the full-disk paths

```
ls -l /dev/disk/by-id/
```

Choose the two **whole-disk** ata- or wwn- links. Do not choose a link ending in -part1 or another partition suffix. The video uses sda and sdc; those letters are not instructions for your machine.

```
D1='/dev/disk/by-id/ata-YOUR_FIRST_DRIVE'
D2='/dev/disk/by-id/ata-YOUR_SECOND_DRIVE'
readlink -f "$D1"
readlink -f "$D2"
lsblk -o NAME,SIZE,MODEL,SERIAL,FSTYPE,MOUNTPOINTS "$D1" "$D2"
```

Stop if the identity or use is uncertain

Both paths must exist and resolve to different spare disks. Exclude boot disks, live VM disks, USB sticks and any disk whose data you still need. These variables exist only in this shell; after reconnecting, set and verify them again.

03 / HEALTH - VIDEO 5:59-7:39

Check before you trust

A SMART PASSED headline is a useful start, rather than a lifetime warranty.

4. Read health and history

```
smartctl -x "$D1"
smartctl -x "$D2"
```

If smartctl is missing, install smartmontools through your configured Proxmox/Debian repositories: **apt install smartmontools**. Review overall health, error logs and self-test history.

For SATA HDDs, inspect Reallocated_Sector_Ct, Current_Pending_Sector and Offline_Uncorrectable. Pending or uncorrectable sectors, failed tests or increasing counts are reasons to reject the drive. CRC errors can indicate a cable problem; check whether the count increases after fixing it.

5. Run an extended self-test on both drives

```
smartctl -t long "$D1"
smartctl -t long "$D2"
```

Each command starts a background test and reports when it should finish. Keep the host powered on and avoid heavy disk activity. Wait for the reported completion time; old disks can take hours.

```
smartctl -l selftest "$D1"
smartctl -l selftest "$D2"
smartctl -x "$D1"
smartctl -x "$D2"
```

Your go / no-go check

Look for the newest extended test completed without error, then review the attributes again. A failed test, unreadable sectors or an unexplained error needs investigation before pool creation. Two questionable drives do not add up to one trustworthy drive.

Keep the model, serial and test result in your notes. Replace suspect drives rather than relying on the mirror to hide their condition.

04 / CLEAR SPARE DISKS - VIDEO 12:49-14:17

The point of no return

Only continue with two verified spare drives whose old contents can be discarded.

This destroys access to existing data

The commands below remove disk signatures and partition tables from both selected disks. They are not secure erasure. Recheck D1 and D2 against the serials from page 2 immediately before running them. Do not use these commands on a disk used by the host or a guest.

6. Preview what is on each drive

```
wipefs -n "$D1"
wipefs -n "$D2"
lsblk -o NAME,SIZE,SERIAL,FSTYPE,MOUNTPOINTS "$D1" "$D2"
pvs
zpool status
```

If a mount, active volume group or pool uses either drive, stop and resolve that use first. Do not force past a busy-device message. Back up any old files you want before clearing the disk.

7. Clear signatures and partition tables

```
wipefs -a "$D1"
wipefs -a "$D2"
sgdisk --zap-all "$D1"
sgdisk --zap-all "$D2"
partprobe "$D1"
partprobe "$D2"
udevadm settle
```

wipefs removes recognised signatures; sgdisk clears GPT and MBR partition-table structures. partprobe asks the kernel to re-read them. If these tools are missing, their packages are util-linux, gdisk and parted.

8. Check the result

```
lsblk -o NAME,SIZE,SERIAL,FSTYPE,MOUNTPOINTS "$D1" "$D2"
wipefs -n "$D1"
wipefs -n "$D2"
```

Expect no old partition tree or whole-disk signatures. If the kernel cannot re-read the table, investigate active use; a planned reboot may be needed. Re-identify the disks afterwards. Do not continue with stale output.

05 / BUILD THE MIRROR - VIDEO 14:18-16:46

Create the pool and folder

A pool combines the drives. A dataset is a managed filesystem inside that pool.

9. Check names and preview the pool

```
zpool list
zpool create -n -o ashift=12 \
  -O compression=lz4 -O atime=off \
  backuppool mirror "$D1" "$D2"
```

Use the name **backuppool** throughout this guide. It must not already exist. The -n preview must show one mirror containing exactly your two disks.

10. Create the pool

```
zpool create -o ashift=12 \
  -O compression=lz4 -O atime=off \
  backuppool mirror "$D1" "$D2"
zpool status -P backuppool
```

Creation overwrites the selected disks

This is a destructive step. Check the preview and serials first. Do not add -f to bypass an unexpected warning. The result should be ONLINE, with both mirror members ONLINE and no known data errors.

ashift=12 selects 4 KiB sector alignment; it is not the file record size. **compression=lz4** enables fast compression. **atime=off** avoids recording every file read. Already compressed backup files may gain little from LZ4.

11. Create the backup dataset

```
zfs create -o mountpoint=/backuppool/backups \
  backuppool/backups
zfs get compression,atime,mounted,mountpoint backuppool/backups
findmnt -M /backuppool/backups
```

Properties inherit from the pool root dataset. Confirm mounted=yes and that findmnt reports source backuppool/backups with type zfs at the exact path. Stop if it is only an ordinary folder on the boot disk.

06 / REGISTER STORAGE - VIDEO 16:47-18:23

Let Proxmox use it

Use Directory storage for backup files held on a ZFS filesystem.

12. Set the path and owning node

```
P=/backuppool/backups
NODE=$(hostname -s)
findmnt -M "$P"
pvesm status
```

NODE must match the Proxmox node name shown in the web interface. If it differs, set NODE to that exact name. Check that the storage ID backup-mirror does not already exist; inspect existing configuration rather than adding a duplicate.

13. Add it disabled, with a mount check

```
pvesm add dir backup-mirror --path "$P" \
  --content backup --disable 1 --create-base-path 0 \
  --is_mountpoint "$P" --nodes "$NODE"
cat /etc/pve/storage.cfg
```

--content backup allows backup archives. **--disable 1** keeps it disabled while you check it. **--create-base-path 0** prevents creating a missing base directory. **--is_mountpoint** keeps storage offline if the dataset is not mounted. **--nodes** restricts it to the host with these local disks.

14. Check and enable

```
findmnt -M "$P"
pvesm set backup-mirror --disable 0
pvesm status --storage backup-mirror
```

In **Datacenter > Storage**, confirm Type=Directory, Content=Backup, Nodes=your host and Path=/backuppool/backups. It should be active. Leave Shared unticked: a node restriction does not turn local disks into network storage.

> ### Why the mount check matters
>
> A folder can still exist when its filesystem is absent. Writing there could fill the Proxmox boot disk. If the storage is inactive, fix the mount instead of removing this protection.

07 / FIRST BACKUP - VIDEO 18:29-18:47

Make a backup you can test

Start with a small, non-critical VM. Success is more useful than a very impressive progress bar.

15. Take a manual backup

Select the VM in Proxmox, open **Backup > Backup now**, choose **backup-mirror**, Compression=**ZSTD** and Mode=**Snapshot**. Start it and read the task log through to **TASK OK**. This backup mode is different from a persistent VM snapshot.

For a VM, the equivalent command below uses an example ID. Replace 101 with your actual test VM ID.

```
vzdump 101 --storage backup-mirror \
  --mode snapshot --compress zstd
pvesm list backup-mirror --content backup
```

Confirm the backup appears with the expected VM ID, date and a plausible size. Guest-agent support can improve consistency; applications may need their own backup procedure. Check whether any disks have backup disabled, and remember passthrough devices are not ordinary backed-up virtual disks.

16. Set a schedule after the restore test

Use **Datacenter > Backup > Add**: select this node, backup-mirror, the guests, a suitable schedule, compression and mode. Set retention to match available capacity. Keep an eye on task results and notification delivery.

> ### Keep another copy
>
> Archives here are local to this host. Copy backups to separate storage or use an independent backup server, ideally with an off-site copy too. A mirror does not protect against losing the whole machine.

Check free space with **zpool list backuppool** and **zfs list backuppool/backups**. Leave headroom, review retention, and avoid pruning the only known-good backup before you have tested another.

08 / RESTORE SAFELY - VIDEO 18:47-20:16

Prove you can get it back

Restore to an unused VM ID. Keep the original VM intact while you test.

17. Choose the archive, ID and target storage

```
qm list
pct list
pvesm status
pvesm list backup-mirror --content backup
```

Pick an unused ID across VMs and containers. Choose a target storage that supports **images** and has enough free space for the restored virtual disks. backup-mirror is backup-only, so it is not the target VM disk storage.

18. Restore a VM copy

In the backup list, select the archive and click Restore. Choose an unused VM ID if offered, the target disk storage and the unique MAC option. Leave Start after restore off. If your interface cannot choose a new ID, use qmrestore.

```
qmrestore /backuppool/backups/dump/EXACT_BACKUP.vma.zst \
  901 --storage YOUR_VM_STORAGE --unique 1
```

Replace the archive path with the exact VM backup filename, 901 with your unused ID and YOUR_VM_STORAGE with the target storage ID. This command is for VM archives; containers use pct restore. Do not add --force or overwrite your original VM for a test.

Isolate before starting

A new MAC address does not change a static IP or stop duplicate automations. Disconnect every network adapter in the restored VM's Hardware settings before its first boot, or use an isolated test network. Review copied passthrough hardware too. Two DHCP servers or two active Home Assistants can make a lively afternoon.

19. Check the recovered guest

Wait for TASK OK, then start the isolated copy and inspect its console, files and application data. Record the result. Shut down the test copy after checking. A completed restore plus an application check is stronger evidence than an archive merely existing.

09 / CARE AND TROUBLESHOOTING - VIDEO 20:17-20:44

Keep the mirror healthy

ZFS checks data checksums. It can repair damage when a valid redundant copy remains.

20. Scrub and inspect

```
zpool scrub backuppool
zpool status -v backuppool
```

The scrub runs in the background. Recheck status later for completion and errors; the first status command does not wait for it. It checks allocated data, while SMART long tests check the drives. Check existing scrub timers before adding another schedule: **systemctl list-timers --all**.

If something goes wrong

Drive missing: power down safely before checking SATA power, data cable and controller settings. Re-run lsblk and verify serials.
Device busy or signatures remain: check mounts, LVM and ZFS ownership; do not bypass the warning with force.
Storage inactive: check zpool status, zfs get mounted,mountpoint backuppool/backups and findmnt -M /backuppool/backups. Only mount the intended dataset after verifying its pool and path.
No space: inspect pool/dataset usage and backup retention. Preserve a verified restore point elsewhere before removing archives.

Restore ID exists: choose another unused ID; do not force an overwrite.
DEGRADED or errors: keep an independent backup current, inspect zpool status -v and SMART, and identify the failing drive by serial. Investigate cables and hardware; clearing counters is not a repair. Replace a failed member with a suitable equal-or-larger disk, then confirm resilver completion.

- ☐ Two healthy mirror members ONLINE; dataset mounted at the correct path.
- ☐ backup-mirror active on the intended node with mount protection.
- ☐ Backup task succeeded and isolated restore checked.
- ☐ Separate backup copy, retention and ongoing health checks arranged.

```
Drive serials / host / last successful restore test
```

10 / REFERENCE DESK

Your command map

Use the page-by-page instructions before making changes. This is a reminder, not a shortcut past the disk checks.

Watch the accompanying WrongWords video

The names used in this guide

D1 / D2: your verified whole-disk IDs.
backuppool: the ZFS pool.
backuppool/backups: the dataset.
/backuppool/backups: its mountpoint.
backup-mirror: the Proxmox Directory storage ID.
NODE: the exact Proxmox node name.

Jump back to the demonstration

5:00 - identify disks; 5:59 - SMART checks; 6:50 - self-tests.
11:17 - recheck disk identity; 12:49 - clear old layout.
14:18 - create the mirror; 15:58 - create the dataset.
16:47 - register Proxmox storage; 18:29 - backup result.
18:47 - restore demonstration; 20:17 - check the mirror.

Official technical references

Proxmox: Directory storage
Proxmox: pvesm options
Proxmox: backup and restore
Proxmox: qmrestore options
OpenZFS: pool creation
OpenZFS: dataset properties
OpenZFS: scrub and repairs
smartctl: health checks and tests
wipefs: signature removal

One final useful habit

Write down which archive you restored, where the separate backup lives and when you tested it. Future you deserves more than a folder called final-final-really-final.