Zmiana nazwy grupy wolumenów i wolumenu logicznego

Poziom: zaawansowany
Dotyczy: RHEL/Rocky 9–10, różnice Debian/Ubuntu i SLES zaznaczone **Czas: ~25 min**

Zmiana nazwy grupy wolumenów albo wolumenu logicznego brzmi jak kosmetyka — sama operacja trwa ułamek sekundy i nie rusza ani bajta danych. Problem w tym, że system startuje z zapamiętaną, starą nazwą wpisaną w trzech różnych miejscach, i jeśli nie zaktualizujesz wszystkich trzech przed restartem, dostajesz serwer, który nie wstaje.

Skróty użyte na tej stronie (14)

Skrót	Co znaczy
BIOS	Basic Input/Output System: oprogramowanie płyty głównej, które startuje komputer
EFI	Extensible Firmware Interface: poprzednik nazwy UEFI; w Proxmoksie dysk efidisk0 trzyma zmienne firmware maszyny z OVMF
GRUB	Grand Unified Bootloader: program, który jako pierwszy startuje po włączeniu komputera i podaje jądru, co ma uruchomić
LV	Logical Volume: wolumen logiczny w LVM, na nim leży system plików
LVM	Logical Volume Manager: warstwa w Linuksie składająca dyski w elastyczne wolumeny
PV	Physical Volume: dysk lub partycja oddana pod LVM; w Kubernetesie PersistentVolume, trwały wolumen
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
SLES	SUSE Linux Enterprise Server: płatna dystrybucja Linuksa firmy SUSE, konkurent Rocky/RHEL
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera
SUSE	nazwa niemieckiej firmy i jej dystrybucji Linuksa, od niemieckiego skrótu Software und System Entwicklung
UEFI	Unified Extensible Firmware Interface: nowoczesne oprogramowanie firmowe płyty głównej albo maszyny wirtualnej, następca starszego BIOS-u, obsługuje między innymi Secure Boot

Skrót	Co znaczy
USB	Universal Serial Bus: standard portów i urządzeń podłączanych do komputera, tu pendrive'y i inne nośniki wymienne
UUID	Universally Unique Identifier: długi, praktycznie niepowtarzalny identyfikator
VG	Volume Group: grupa wolumenów w LVM, pula złożona z dysków

Fiszka

TYLKO ODCZYT

Wypisanie grup wolumenów w systemie: vgs

```
vgs
```

Pokazujemy wszystkie grupy wolumenów widoczne w systemie, razem z liczbą wolumenów fizycznych i logicznych, jakie zawierają. Punkt startowy przed jakąkolwiek operacją na LVM — trzeba wiedzieć, jak grupa nazywa się *teraz*.

Przykład:

```
  VG   #PV #LV #SN Attr   VSize   VFree
  rl     1   3   0 wz--n- <99,00g    0
```

LVM = Logical Volume Manager: warstwa w Linuksie składająca dyski w elastyczne wolumeny · **vgs** = volume groups: pokaż listę grup wolumenów LVM

TYLKO ODCZYT

Wypisanie wolumenów logicznych: lvs

```
lvs
```

To samo co `vgs`, ale na poziomie pojedynczych wolumenów logicznych — nazwa, grupa, do której należą, i rozmiar.

Przykład:

```
  LV   VG Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  root rl -wi-ao----  30,00g
  home rl -wi-ao----  50,00g
  swap rl -wi-ao----  19,00g
```

lvs = logical volumes: pokaż listę wolumenów logicznych LVM

TYLKO ODCZYT

Zmiana nazwy grupy wolumenów: vgrename

```
vgrename rl vgOS
```

Operacja czysto metadanowa — LVM zmienia wpis w nagłówku grupy i przelicza wewnętrzne odwołania. Trwa ułamek sekundy, dane leżą dokładnie tam, gdzie leżały.

Przykład:

```
  Volume group "rl" successfully renamed to "vgOS"
```

LVM = Logical Volume Manager: warstwa w Linuksie składająca dyski w elastyczne wolumeny

TYLKO ODCZYT

Zmiana nazwy wolumenu logicznego: lvrename

```
lvrename vgOS root lvOS
```

Ta sama zasada co przy `vgrename`: zmienia się etykieta, nie zawartość. Po tej komendzie węzeł `/dev/mapper/vgOS-root` przestaje istnieć, a w jego miejsce pojawia się `/dev/mapper/vgOS-lvOS`.

Przykład:

```
  Renamed "root" to "lvOS" in volume group "vgOS"
```

ZMIENIA SYSTEM

Poprawienie `/etc/fstab`

```
sed -i 's#/dev/mapper/rl-root#/dev/mapper/vgOS-lvOS#;
        s#/dev/mapper/rl-home#/dev/mapper/vgHome-lvHome#;
        s#/dev/mapper/rl-swap#/dev/mapper/vgOS-lvSwap#' /etc/fstab
```

`/etc/fstab` odwołuje się do wolumenów po ścieżce `/dev/mapper/<vg>-<lv>`, więc każdy wpis ze starą nazwą trzeba przepisać. Jeśli wpisy w `fstab` używają UUID-ów zamiast ścieżek `/dev/mapper`, ten krok w ogóle nie jest potrzebny — kolejny argument za montowaniem po UUID, opisany na osobnej stronie o `mount` i `fstab`.

UUID = Universally Unique Identifier: długi, praktycznie niepowtarzalny identyfikator · **sed** = stream editor: edytor strumienia tekstu

ZMIENIA SYSTEM

Poprawienie `/etc/default/grub`: rd.lvm.lv=

```
sed -i 's#rl/root#vgOS/lvOS#g; s#rl/swap#vgOS/lvSwap#g' /etc/default/grub
```

`GRUB_CMDLINE_LINUX` w tym pliku zawiera parametr `rd.lvm.lv=`, który mówi wczesnemu środowisku rozruchowemu (initramfs), które konkretnie wolumeny logiczne aktywować, zanim jeszcze `/etc/fstab` zostanie odczytane. Stara nazwa w tym parametrze to najczęstsza przyczyna nieudanego bootu po zmianie nazwy VG.

Przykład:

```
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=rl/root rd.lvm.lv=rl/swap rhgb quiet"
```

VG = Volume Group: grupa wolumenów w LVM, pula złożona z dysków · **sed** = stream editor: edytor strumienia tekstu

TYLKO ODCZYT

Przebudowa configu na BIOS: grub2-mkconfig -o

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

Na maszynie startującej w trybie BIOS/Legacy plik startowy GRUB-a leży w `/boot/grub2/grub.cfg`. Na Debianie/Ubuntu odpowiednikiem jest `update-grub` (który sam wywołuje `grub-mkconfig -o /boot/grub/grub.cfg` — inna ścieżka, inna nazwa katalogu).

Przykład:

```
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.14.0-427.el9.x86_64
Found initrd image: /boot/initramfs-5.14.0-427.el9.x86_64.img
done
```

BIOS = Basic Input/Output System: oprogramowanie płyty głównej, które startuje komputer

TYLKO ODCZYT

Przebudowa configu na UEFI: grub2-mkconfig -o

```
grub2-mkconfig -o /boot/efi/EFI/rocky/grub.cfg
```

Na maszynie UEFI plik startowy leży w zupełnie innym miejscu, na partycji EFI System Partition — ścieżka zawiera nazwę dystrybucji (`rocky`, `redhat`, `centos`...). Przebudowanie configu pod złym trybem (BIOS zamiast UEFI albo odwrotnie) nie da żadnego błędu — po prostu poprawimy plik, którego bootloader w ogóle nie czyta. Sprawdzenie, w jakim trybie wystartował system:

```
[ -d /sys/firmware/efi ] && echo UEFI || echo BIOS
```

UEFI = Unified Extensible Firmware Interface: nowoczesne oprogramowanie firmowe płyty głównej albo maszyny wirtualnej, następca starszego BIOS-u, obsługuje między innymi Secure Boot · **EFI** = Extensible Firmware Interface: poprzednik nazwy UEFI; w Proxmoksie dysk efidisk0 trzyma zmienne firmware maszyny z OVMF · **BIOS** = Basic Input/Output System: oprogramowanie płyty głównej, które startuje komputer

TYLKO ODCZYT

Przebudowa initramfs na RHEL i SUSE: dracut -v -f

```
dracut -v -f
```

Bez argumentów przebudowujemy initramfs dla aktualnie uruchomionego jądra, nadpisując istniejący plik (`-f` — force). `-v` włącza tryb gadatliwy, przydatny, gdy coś idzie nie tak. Na nowszych wersjach SUSE (SLES 12 i wyżej) ramdisk generuje się tą samą komendą — starszy `mkinitrd` odszedł do historii razem z klasycznym SUSE 11.

RHEL = Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa · **SUSE** = nazwa niemieckiej firmy i jej dystrybucji Linuksa, od niemieckiego skrótu Software und System Entwicklung · **SLES** = SUSE Linux Enterprise Server: płatna dystrybucja Linuksa firmy SUSE, konkurent Rocky/RHEL

□□ ZMIENIA SYSTEM

Przebudowa initramfs na Debianie/Ubuntu: update-initramfs -u

```
update-initramfs -u -k all
```

Odpowiednik `dracut -v -f` w świecie Debiana. `-u` znaczy update, a `-k all` każe przebudować initramfs dla wszystkich zainstalowanych wersji jądra, nie tylko dla bieżącej — ważne, jeśli w `/boot` leży więcej niż jedno jądro.

□□ TYLKO ODCZYT

Weryfikacja zawartości initramfs przed restartem: lsinitrd | grep

```
lsinitrd /boot/initramfs-$(uname -r).img | grep -i lvm
```

`lsinitrd` rozpakowuje initramfs do podglądu, bez modyfikowania niczego. Szukamy w nim śladów nowej nazwy VG/LV w plikach konfiguracyjnych LVM zaszytych wewnątrz obrazu — jeśli grep nadal zwraca starą nazwę, initramfs nie został poprawnie przebudowany.

Przykład:

```
-rw-r--r--   1 root     root          138 lvm/lvm.conf
-rw-r--r--   1 root     root           41 lvm/cache/.cache
```

□□ **VG** = Volume Group: grupa wolumenów w LVM, pula złożona z dysków · **LVM** = Logical Volume Manager: warstwa w Linuksie składająca dyski w elastyczne wolumeny

□□ TYLKO ODCZYT

Weryfikacja przed restartem — trzy miejsca naraz

```
grep -c "rl-\|rl/" /etc/fstab /boot/grub2/grub.cfg
lsinitrd /boot/initramfs-$(uname -r).img | grep -c "rl/"
```

Zero wyników starej nazwy we wszystkich trzech miejscach to jedyny sygnał, że restart jest bezpieczny. Jedna zapomniana litera w `sed` i jeden z trzech plików nadal wskazuje na nieistniejącą już grupę wolumenów.

Przykład:

```
0
0
0
```

grep = global regular expression print: szukaj wzorca w tekście

TYLKO ODCZYT

Podsumowanie kolejności

Krok	Co robi	Dlaczego w tej kolejności
`vgrename`, `lvrename`	zmienia nazwę w metadanych LVM	musi być pierwsze — kolejne kroki odwołują się już do nowej nazwy
poprawka `/etc/fstab`	aktualizuje ścieżki montowania	systemd czyta ten plik po starcie jądra
poprawka `/etc/default/grub`	aktualizuje `rd.lvm.lv=`, `root=`	wpływa na to, co trafi do configu GRUB-a w następnym kroku
`grub2-mkconfig` (BIOS i/lub UEFI)	przepisuje faktyczny plik startowy bootloadera	musi być po poprawce `/etc/default/grub`, inaczej przepisze stare wartości
`dracut -v -f` / `update-initramfs -u`	przebudowuje initramfs	initramfs ma zaszyte metadane LVM z chwili budowy — muszą być aktualne
weryfikacja (`grep`, `lsinitrd`)	potwierdza brak starej nazwy	ostatni krok przed restartem, nie po nim

Uważaj

- Kolejność ma znaczenie: `grub2-mkconfig` uruchomione *przed* poprawką `/etc/default/grub` wygeneruje plik ze starą nazwą, mimo że sam VG już jest przemianowany.
- Dla wolumenu z danymi (nie root, nie boot, nie swap) wystarczy `vgrename`/`lvrename` i poprawka `/etc/fstab` — kroki z GRUB-em i initramfs dotyczą wyłącznie wolumenów potrzebnych do wystartowania jądra.
- Nigdy nie restartujemy między krokami „na próbę” — dopóki wszystkie trzy miejsca nie są spójne, każdy restart to loteria.

BIOS = Basic Input/Output System: oprogramowanie płyty głównej, które startuje komputer · **UEFI** = Unified Extensible Firmware Interface: nowoczesne oprogramowanie firmowe płyty głównej albo maszyny wirtualnej, następca starszego BIOS-u, obsługuje między innymi Secure Boot · **LVM** = Logical Volume Manager: warstwa w Linuksie składająca dyski w elastyczne wolumeny · **VG** =

Volume Group: grupa wolumenów w LVM, pula złożona z dysków

Z życia wzięte

Z ŻYCIA WZIĘTE

Poniedziałek po Patch Tuesday. Ktoś zmienił nazwę grupy wolumenów na starym serwerze płacowym syb01, nie sprawdzając wcześniej, gdzie ta nazwa jest jeszcze zapisana. System nie wstaje po restarcie, bo GRUB i initramfs wciąż szukają dysku pod starą nazwą, a poprawka wymaga wejścia w tryb ratunkowy z konsoli, nie przez SSH.

Po co to istnieje

Pojęcie: wolumen fizyczny, grupa wolumenów, wolumen logiczny (PV, VG, LV). LVM (Logical Volume Manager) to warstwa pośrednia między fizycznymi dyskami a systemami plików. Dysk albo partycja staje się wolumenem fizycznym (PV) po `pvcreate`. Kilka PV można połączyć w jedną pulę miejsca zwaną grupą wolumenów (VG) — od tego momentu granice fizycznych dysków przestają mieć znaczenie dla administratora. Z tej puli wykrawa się wolumeny logiczne (LV), czyli odpowiedniki partycji, ale elastyczne: da się je powiększyć, pomniejszyć, przenieść między dyskami bez odmontowywania systemu plików, który na nich leży. To właśnie ta elastyczność jest powodem, dla którego niemal każdy serwer produkcyjny w tej książce stoi na LVM, a nie na gołych partycjach.

Nazwy VG i LV nie są magiczne — to zwykłe etykiety tekstowe w metadanych LVM. Powód, żeby je zmieniać, pojawia się częściej, niż mogłoby się wydawać: konsolidacja po migracji fizyczny-do-wirtualnego (P2V), ujednolicenie nazewnictwa po przejęciu serwera od innego zespołu, albo sprzątanie po latach, gdy domyślne `rl` nie pasuje już do standardu `vg<Przeznaczenie>` wprowadzonego później. Sama operacja jest trywialna — trudność bierze się z tego, że nazwa VG/LV rozrasta się po całym systemie, w plikach czytanych na długo przed tym, zanim jądro w ogóle dotrze do `/etc/fstab`.

Analogia

Wyobraź sobie, że miasto zmienia nazwę ulicy — z ulicy Kolejowej na ulicę Nowohucką. Urząd miasta aktualizuje rejestr w kilka minut: od teraz ta ulica *oficjalnie* nazywa się inaczej. To jest dokładnie to, co robi `vgrename`.

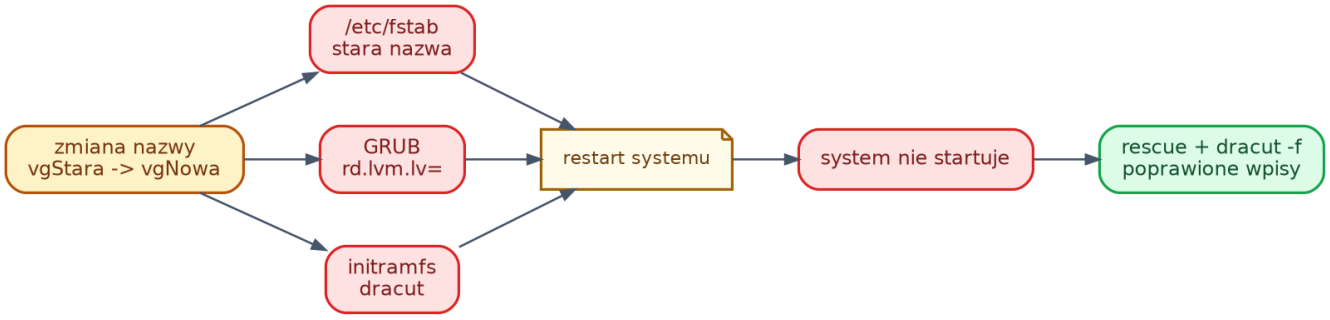
Problem w tym, że listonosz roznoszący pocztę korzysta z własnej, wydrukowanej mapy sprzed zmiany. Będzie nadal szukał ulicy Kolejowej i jej nie znajdzie, mimo że budynki stoją tam, gdzie stały. Listonosz to bootloader i initramfs: mają własną, „wydrukowaną” kopię starej nazwy zaszytą w konfiguracji i będą jej szukać, dopóki ktoś świadomie nie zaktualizuje ich mapy — czyli `/etc/default/grub` i initramfs.

Różnica między wolumenem z danymi a wolumenem roota jest jak różnica między pomyłką w adresie sklepu spożywczego a pomyłką w adresie samego urzędu pocztowego. W pierwszym przypadku klienci dziś nie dostaną paczki — irytujące, ale poczta jako całość działa. W drugim cała poczta w mieście przestaje działać, bo stamtąd wyjeżdżają wszystkie trasy. Dokładnie to dzieje się przy pomyłce z wolumenem roota: system w ogóle nie wstaje, bo jądro nie potrafi znaleźć własnego korzenia.

Jak to działa

Rozwiń: jak to działa pod spodem

Kolejność, w jakiej Linux odnajduje swój system plików root, tłumaczy, dlaczego trzy pliki muszą być spójne, zanim zrestartujesz maszynę:



Zmiana nazwy VG lub LV dotyka fstab, GRUB i initramfs naraz, więc restart bez poprawek kończy się brakiem rozruchu. (kliknięcie otwiera pełny rozmiar)

1. GRUB odczytuje `grub.cfg`, znajduje wpis jądra i initramfs, i przekazuje jądru parametry z linii `linux` — w tym `root=` i `rd.lvm.lv=`.
2. Jądro rozpakowuje initramfs do pamięci i uruchamia w nim `dracut` (albo mechanizm `initramfs-tools` na Debianie) jako wczesne środowisko userspace.
3. `dracut` czyta przekazany parametr `rd.lvm.lv=vgOS/lvOS` i aktywuje **tylko** wskazany wolumen logiczny — nie każdy VG widoczny w systemie. To ograniczenie jest celowe: bez niego dracut próbowałby aktywować wszystkie grupy wolumenów, co bywa

problemem przy zduplikowanych UUID-ach PV (np. po sklonowaniu dysku).
4. Po aktywacji LV, jądro montuje go jako system plików root i przekazuje sterowanie do prawdziwego `/sbin/init` (systemd) na dysku docelowym.
5. Dopiero systemd, uruchomiony już z właściwego roota, czyta `/etc/fstab` i montuje pozostałe systemy plików — `/home`, `/var`, swap.

Initramfs nie czyta `/etc/fstab` ani `/etc/default/grub` — ma własną, wcześniej zapisaną kopię konfiguracji LVM, spakowaną do obrazu w momencie budowy. Jeśli zmienisz nazwę VG i nie przebudujesz initramfs, ta wewnętrzna kopia nadal „zna" tylko starą nazwę — nawet jeśli `rd.lvm.lv=` w GRUB-ie już wskazuje na nową. Stąd dwa osobne kroki: aktualizacja parametru w GRUB-ie i osobno przebudowa initramfs — jeden bez drugiego nie wystarczy.

Dla wolumenu z danymi montowanego już po starcie systemd sytuacja jest łagodniejsza: systemd generuje jednostkę `.mount` już po tym, jak root jest zamontowany i system żyje. Błąd w tym miejscu kończy się nieudanym montowaniem jednego katalogu (`Failed to mount /home`), a nie brakiem bootu w ogóle.

Krok po kroku

Poniższa kolejność dotyczy najtrudniejszego przypadku — zmiany nazwy VG/LV zawierającego **root**. Dla wolumenu z danymi wystarczą kroki 1–5, bez GRUB-a i initramfs.

1. Zrób kopię plików konfiguracyjnych.

```
cp /etc/fstab /etc/fstab.bak
cp /etc/default/grub /etc/default/grub.bak
```

Jeśli coś pójdzie nie tak przy edycji `sed`-em, wracasz do punktu wyjścia jedną komendą `cp` w drugą stronę.

2. Sprawdź aktualny stan.

```
vgs
lvs -o +devices
cat /etc/fstab
```

Zanotuj dokładne nazwy — literówka na tym etapie propaguje się przez wszystkie kolejne kroki.

3. Zmień nazwę grupy wolumenów.

```
vgrename rl vgOS
```

```
  Volume group "rl" successfully renamed to "vgOS"
```

4. Zmień nazwę wolumenu logicznego.

```
lvrename vgOS root lvOS
```

```
  Renamed "root" to "lvOS" in volume group "vgOS"
```

5. Zaktualizuj `/etc/fstab`.

```
sed -i 's#/dev/mapper/rl-root#/dev/mapper/vgOS-lvOS#g' /etc/fstab
grep vgOS /etc/fstab
```

6. Zaktualizuj `/etc/default/grub`.

```
sed -i 's#rl/root#vgOS/lvOS#g' /etc/default/grub
grep rd.lvm.lv /etc/default/grub
```

7. Przebuduj `grub.cfg` we właściwej ścieżce.

```
[ -d /sys/firmware/efi ] && grub2-mkconfig -o /boot/efi/EFI/rocky/grub.cfg \
                         || grub2-mkconfig -o /boot/grub2/grub.cfg
```

8. Przebuduj initramfs.

```
dracut -v -f
```

9. Zweryfikuj wszystkie trzy miejsca, zanim dotkniesz `reboot`.

```
grep -c "rl-\|rl/" /etc/fstab /boot/grub2/grub.cfg
lsinitrd /boot/initramfs-$(uname -r).img | grep -c "rl/"
```

```
0
0
0
```

Trzy zera. Dopiero teraz restart jest bezpieczny.

Odzyskiwanie z trybu ratunkowego

Jeśli mimo weryfikacji system nie wstaje — albo weryfikacja została pominięta i już jesteś po restarcie — procedura ratunkowa wygląda tak:

1. **Uruchom system z nośnika instalacyjnego** w trybie ratunkowym (Rescue a Rocky Linux system / Rescue an installed system) albo z dowolnego live USB z narzędziami LVM.
2. **Sprawdź prawdziwy stan LVM** — rename już się wykonał przed restartem, więc dane i metadane są spójne, nawet jeśli konfiguracja startowa nie jest:

``bash vgchange -ay vgs lvs -o +devices` Wynik pokaże Ci *aktualną*, już zmienioną nazwę — to jest źródło prawdy, a nie stary grub.cfg`.

1. **Zamontuj system plików root ręcznie i wejdź w `chroot`:**

``bash mount /dev/mapper/vgOS-lvOS /mnt/sysimage mount /dev/mapper/vgOS-lvBoot /mnt/sysimage/boot # jeśli /boot jest osobnym LV for d in dev proc sys run; do mount --bind /$d /mnt/sysimage/$d; done chroot /mnt/sysimage``

1. **Popraw `/etc/fstab` i `/etc/default/grub`** dokładnie tak, jak w krokach 5–6 powyżej — tym razem wewnątrz `chroot`.
2. **Przebuduj `grub.cfg` i initramfs** wewnątrz `chroot`, tymi samymi komendami co w krokach 7–8.
3. **Wyjdź z `chroot`, odmontuj i zrestartuj:**

``bash exit umount -R /mnt/sysimage reboot``

Dla wolumenu z danymi (nie root, nie boot) trybu ratunkowego zwykle nie potrzebujesz — system wstaje normalnie, tylko jedna jednostka `.mount` się nie uruchamia, i naprawiasz to z działającego systemu. Dla roota i swapa procedura jest zauważalnie trudniejsza: swap wymaga dodatkowo poprawnej wartości `resume=` w linii jądra (jeśli używana jest hibernacja) i regeneracji initramfs, bo moduł `resume` też ma zaszytą nazwę urządzenia — pominięcie tego nie blokuje bootu, ale psuje wybudzanie z hibernacji w sposób trudny do zdiagnozowania później.

Częste błędy

Initramfs nie przebudowany — system wpada w powłokę awaryjną dracut.

```
Warning: dracut-initqueue timeout - starting timeout scripts
Warning: /dev/mapper/rl-root does not exist
Generating "/run/initramfs/rdsosreport.txt"


dracut:/#
```

Klasyczny objaw pominięcia `dracut -v -f`. GRUB poprawnie przekazał nową nazwę w `rd.lvm.lv=`, ale initramfs ma zaszytą starą konfigurację LVM. Naprawa: rozruch ratunkowy, `chroot`, `dracut -v -`

`f`, restart.

Zapomniany parametr `rd.lvm.lv=` w GRUB-ie.

```
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(0,0)
```

Initramfs jest już poprawny, ale `/etc/default/grub` nadal wskazuje starą nazwę w `rd.lvm.lv=` — dracut próbuje aktywować wolumen, który już nie istnieje pod tą nazwą. Naprawa: rozruch ratunkowy, poprawka `/etc/default/grub`, ponowne `grub2-mkconfig`.

`grub2-mkconfig` uruchomiony pod złą ścieżką na maszynie UEFI.

Polecenie kończy się sukcesem, `/boot/grub2/grub.cfg` zostaje zaktualizowany — tyle że system UEFI go nie czyta. Bootloader nadal korzysta ze starego `/boot/efi/EFI/rocky/grub.cfg`, więc po restarcie wraca ten sam błąd. Sprawdzenie trybu firmware (`[ -d /sys/firmware/efi ]`) przed tym krokiem eliminuje pomyłkę.

Zapomniany wpis w `/etc/fstab` dla wolumenu innego niż root.

```
[ TIME ] Timed out waiting for device /dev/mapper/vgHome-lvHome.
[DEPEND] Dependency failed for /home.
Give root password for maintenance
(or press Control-D to continue):
```

Root i initramfs są już zaktualizowane, system startuje — ale `/etc/fstab` nadal ma starą nazwę dla `/home`, więc systemd czeka na urządzenie, które nigdy się nie pojawi. Naprawa: zalogować się hasłem roota z trybu konserwacji, poprawić `/etc/fstab`, `systemctl daemon-reload`, `mount -a`.

Restart „na sprawdzenie”, zanim wszystkie trzy pliki zostały zaktualizowane.

Najczęstsza przyczyna wezwań do trybu ratunkowego nie jest techniczna, tylko proceduralna: administrator zmienia `vgrename`/`lvrename`, od razu restartuje „żeby sprawdzić, czy działa”, i dopiero po nieudanym boocie zabiera się za GRUB i initramfs. Kolejność z *Krok po kroku* nie jest przypadkowa — restart jest ostatnim krokiem, nie środkowym.

□□ Ćwiczenia

> Ćwiczenia korzystają z tymczasowych dysków utworzonych z plików przez `losetup`, więc nic nie dotyka prawdziwego systemu ani prawdziwego rozruchu. Operacje na LVM wymagają uprawnień administratora (`sudo`).

Ćwiczenie 1 — bezpieczna zmiana nazwy na testowej grupie wolumenów

```
cd /tmp
dd if=/dev/zero of=disk1.img bs=1M count=64
sudo losetup -fP disk1.img
LOOP=$(losetup -j disk1.img | cut -d: -f1)
sudo pvcreate "$LOOP"
sudo vgcreate vgTest "$LOOP"
sudo lvcreate -l 100%FREE -n lvTest vgTest
sudo lvs -o +devices | grep vgTest
```

Teraz zmień nazwy i zweryfikuj efekt:

```
sudo vgrename vgTest vgTestNew
sudo lvrename vgTestNew lvTest lvTestNew
sudo lvs -o +devices | grep vgTestNew
```

Co dokładnie zmieniło się w kolumnie `Devices` po zmianie nazwy — sam napis, czy też fizyczne położenie danych na dysku?

Odpowiedź

Zmienia się wyłącznie napis — kolumna `Devices` pokazuje teraz `/dev/vgTestNew/lvTestNew(0)` zamiast `/dev/vgTest/lvTest(0)`, ale wskazuje na dokładnie ten sam obszar tego samego urządzenia `$LOOP`. `vgrename`/`lvrename` nie przenoszą ani jednego ekstentu — to operacja czysto metadanowa, stąd trwa poniżej sekundy niezależnie od rozmiaru wolumenu.

Sprzątanie:

```
sudo lvremove -f vgTestNew/lvTestNew
sudo vgremove vgTestNew
sudo pvremove "$LOOP"
sudo losetup -d "$LOOP"
rm disk1.img
```

Ćwiczenie 2 — wykrywanie niespójności w plikach startowych

Bez prawdziwego GRUB-a i initramfs, symulujemy sam mechanizm sprawdzania:

```
cd /tmp
cat > fstab-test <<'EOF'
/dev/mapper/vgOS-lvOS   /       xfs   defaults   0 0
/dev/mapper/vgHome-lvHome /home   xfs   defaults   0 0
EOF
cat > grub-test <<'EOF'
GRUB_CMDLINE_LINUX="rd.lvm.lv=vgOS/lvOS rd.lvm.lv=vgOS/lvHome crashkernel=auto"
EOF

# Symulacja zmiany nazwy: vgOS -> vgOS, lvOS -> lvOS
sed -i 's#vgOS-lvOS#vgOS-lvOS#; s#vgHome-lvHome#vgHome-lvHome#' fstab-test
sed -i 's#vgOS/lvOS#vgOS/lvOS#; s#vgOS/lvHome#vgOS/lvHome#' grub-test

grep -c vgOS fstab-test grub-test
```

Jaki wynik oznacza, że oba pliki są gotowe do restartu, a jaki — że coś zostało pominięte?

Odpowiedź

```
fstab-test:0
grub-test:0
```

Dwa zera oznaczają brak jakiegokolwiek śladu starej nazwy — bezpieczny restart. Gdyby `grep -c` zwrócił liczbę większą od zera dla któregoś pliku, oznaczałoby to niepełną podmianę (np. literówkę w `sed`, przez co jedno z trzech wystąpień starej nazwy przetrwało) i restart skończyłby się bootem w trybie awaryjnym.

Sprzątanie: `rm fstab-test grub-test`.

Pytania kontrolne

1. Dlaczego samo `vgrename` i `lvrename` wystarcza dla wolumenu z danymi, ale nie wystarcza dla wolumenu roota?

Odpowiedź

Wolumen z danymi jest montowany przez systemd na podstawie `/etc/fstab`, już po tym, jak jądro wystartowało z działającego roota. Błąd w jego nazwie kończy się nieudanym montowaniem jednej jednostki `.mount` — system startuje, tylko ten jeden katalog jest pusty. Wolumen root jest aktywowany dużo wcześniej, przez initramfs sterowany parametrem `rd.lvm.lv=` z linii poleceń jądra, zanim `/etc/fstab` w ogóle zostanie odczytane. Błąd w jego nazwie oznacza, że jądro nigdy nie znajdzie systemu plików root i zatrzyma się na starcie — stąd konieczność aktualizacji GRUB-a i initramfs tylko dla wolumenów biorących udział w rozruchu.

2. Zaktualizowałeś initramfs i `/etc/fstab`, ale zapomniałeś zaktualizować `rd.lvm.lv=` w `/etc/default/grub`. Co się stanie po restarcie?

Odpowiedź

Initramfs ma już poprawną, przebudowaną konfigurację LVM — potrafiłby aktywować nowo nazwany wolumen. Problem w tym, że dracut w ogóle nie spróbuje, bo parametr `rd.lvm.lv=` przekazany przez jądro nadal wskazuje starą nazwę, której już nie ma. Efekt: `dracut-initqueue timeout` i powłoka awaryjna, mimo że sam obraz initramfs jest technicznie aktualny. To pokazuje, że trzy miejsca (fstab, grub, initramfs) trzeba traktować jako jeden spójny zestaw, nie jako listę niezależnych zadań.

3. Jesteś w trybie ratunkowym po nieudanym boocie. Skąd wiesz, czy grupa wolumenów faktycznie ma już nową nazwę, czy operacja `vgrename` się nie powiodła?

Odpowiedź

Uruchamiasz `vgchange -ay`, a potem `vgs`/`lvs` z poziomu środowiska ratunkowego. Metadane LVM są źródłem prawdy — jeśli `vgrename` zakończyło się sukcesem (a zwykle tak jest, bo to prosta operacja metadanowa), `vgs` pokaże już nową nazwę, niezależnie od tego, co jest wpisane w `grub.cfg` czy initramfs. Nieudany boot w tym scenariuszu prawie zawsze oznacza problem z *konfiguracją startową*, nie z samym LVM.

4. Dlaczego procedura dla wolumenu ze swapem jest trudniejsza niż dla zwykłego wolumenu z danymi, mimo że swap też „tylko” się montuje?

Odpowiedź

Swap pojawia się nie tylko w `/etc/fstab`, ale — jeśli system korzysta z hibernacji — także jako parametr `resume=` w linii poleceń jądra, wykorzystywany przez moduł `resume` zaszyty w initramfs do odnalezienia obrazu hibernacji przy starcie. Zmiana nazwy LV ze swapem wymaga więc zaktualizowania obu miejsc i przebudowania initramfs, tak jak w przypadku roota — pominięcie `resume=` nie blokuje bootu, ale cicho psuje wybudzanie z hibernacji, co bywa trudniejsze do zdiagnozowania niż zwykły brak swapu.

Powiązane: Podstawy LVM: PV, VG, LV · pvmove — przenoszenie danych między dyskami · Montowanie: mount, /etc/fstab, zasoby sieciowe

Źródła: `man 8 vgrename` · `man 8 lvrename` · `man 8 dracut` · `man 5 dracut.cmdline` · How to rename the volume group for root and swap — thegeekdiary.com · How to PROPERLY rename Logical Volume — golinuxcloud.com

---

Revision #16
Created 2026-09-17 21:43:06 UTC by Tomasz Konieczny
Updated 2026-10-03 19:08:48 UTC by Tomasz Konieczny