

Jak ustawić LVM, jak robić snapshoty oraz automatycznie powiększać LV, czyli małe howto

Article Number: 7 | Rating: Unrated | Last Updated: Mon, May 21, 2018 9:43 PM

1. Co to jest i do czego służy

LVM (Logical Volume Manager) pozwala na grupowanie urządzeń w wirtualne Volume Group (VG, nie lubię polskich tłumaczeń ;p) z których potem można tworzyć Logical Volume (LV) czyli taki wirtualny odpowiednik partycji dyskowych.

1.1 No dobra, ale po co mi coś takiego ?

LV różni się od fizycznych partycji tym że po pierwsze, jeden LV może rozciągać się po paru fizycznych dyskach, oprócz tego obsługuje zmianę rozmiaru w locie, więc jeżeli kiedyś zabraknie Ci miejsca możesz po prostu dorzucić drugi dysk, dodać go do VG i powiększyć LV o dodatkowe miejsce.

1.2 Czym to się różni od RAID i dlaczego nie użyć RAID zamiast tego

RAID Pozwala na zgrupowanie dysków tak żeby były widziane przez system jako jeden, ale nie pozwala na takie rzeczy jak zmienianie rozmiaru partycji w locie, snapshoting (wiecej o tym później) czy zwiększanie rozmiaru dostępnego storage.

Opócz tego RAID świetnie współgra z LVM, wystarczy użyć RAID-u jako bazę do LVMowego VG i można cieszyć się bezpiecznym i elastycznym storage.

2. Ok, chce mieć u siebie LVM, jak zacząć ?

Pokażę to sposób instalacji i konfiguracji LVM pod debianem, pod innymi distro powinno to wyglądać podobnie, po dokładniejsze instrukcje instalacji zapraszam do [LVM HOWTO](#)

2.1 Instalacja niezbędnego softu

Narzędzia userspace:

```
aptitude install lvm2 reiserfsprogs
```

Kernel:

W kernelu musisz mieć włączone (zalecam wkompiłowane w kernel):

```
Device Drivers -> Multiple devices driver support (RAID and LVM) ->  
Device mapper support  
[*] Snapshot target  
[*] Mirror target  
[*] DM uevents
```

2.1 Mój pierwszy LVM

Na początku musimy przygotować partycje pod LVM:

```
pvcreate /dev/sdc1
```

Następnie tworzymy nowy VG:

```
vgcreate testvg /dev/sdc1
```

Mozna podać parę PV (Physical volume) przy tworzeniu, należy tylko pamiętać o zainicjowaniu ich przez pvcreate

Teraz możemy już utworzyć pierwszy LV

```
lvcreate -L 10G -n muzyka testvg
```

Tworzy nam LV o nazwie muzyka w LV testvg o rozmiarze 10G

Urządzenie które reprezentuje ten LV to /dev/mapper/testvg-muzyka (na marginesie debian tworzy też /dev/testvg/muzyka nie wiem jak inne distro) więc teraz możemy stworzyć i zamontować partycje:

```
mkreiserfs /dev/mapper/testvg-muzyka
mount /dev/mapper/testvg-muzyka /var/music
```

Analogicznie można tworzyć dodatkowe LV. Polecam reiserfs z powodu tego że można go łatwo resizować w locie ;].

2.2 Zabrakło mi miejsca na jakimś LVM, co robić ?

Możesz zwiększyć wielkość LV i systemu plików na nim. Najpierw sprawdź czy masz miejsce na VG:

```
vgdisplay testvg
- Volume group -
VG Name                testvg
System ID
Format                 lvm2
Metadata Areas         1
Metadata Sequence No  152
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 2
Open LV                 1
Max PV                 0
Cur PV                 1
Act PV                 1
VG Size                 180,25 GB
PE Size                 128,00 MB
Total PE                1442
Alloc PE / Size        504 / 63,00 GB
Free PE / Size          938 / 117,25 GB
VG UUID                 WNqUPq-g2x2-BMwg-83Pq-D26a-XpYV-qTZNAF
```

Jak widać mamy jeszcze trochę wolnego miejsca. Teraz tylko

```
lvresize -L +10G /dev/mapper/testvg-muzyka
resize_reiserfs -s +10G /dev/mapper/testvg-muzyka
```

i voila, mamy więcej miejsca na nasze MP3 ;]

2.3 Cool ale mój dysk jest pełen, jak dodać nowe urządzenie do LVM?

```
pvcreate /dev/sdd  
vgextend testvg /dev/sdd
```

I tyle. Teraz tylko powiększyć twoje LV.

3 Zaawansowane zastosowania

3.1 Snapshot

Zacznę może od końca. Wyobraź sobie że masz do zrobienia backup jakiegoś dużego “czegoś”. Nieważne czy to strona, czy jakiś fileshare, masz 50GB do zbackupowania. Nis prostrzego, piszesz jakiś skrypt lub używasz jakiegoś softu typu Bacula, 2h i backup zrobiony. Jest tylko jeden problem.

Pliki zmieniają się podczas backupowania. Jakiś soft zmienia właśnie coś w pierwszej części pliku podczas gdy program właśnie backupuje drugą część tego pliku i w rezultacie plik się popsuł. Jak temu zaradzić ?

Snapshot w LVM jest to obraz systemu plików z danego momentu. Jak to działa ? Przydzielasz snapshotowi powiedzmy 15GB i dopóki na LV-matce nie zmieni się 15GB (przy czym wielokrotne zmiany w tym samym miejscu dysku liczone są jako jedna zmiana) to snapshot będzie pokazywał stan z danego czasu

Działa to tak że od zrobienia snapshotu każdy zapis na LV-matce powoduje to że stara zawartość kopiowana jest na LV-snapshot.

Co to nam daje ? To że można zrobić snapshot np. bazy danych (choć lepszą opcją jest backupowanie bazy przez zrobienie jej dumpa na dysk) o 23:20, potem odpalic na nim backup i ma się pewność że przez najbliższe 30 min podczas robienia backupu nic się na tym nie zmieni, tworząc działający backup

Ułatwia też np. takie rzeczy jak testowanie nowych appów, czy update'ów. Tworzymy snapshot naszego appa, potem robimy upgrade na snapshotcie, sprawdzamy czy działa, usuwamy snapshot.

No to jak to zrobić ?

```
lvcreate -size 10G -snapshot -name appsnap /dev/mapper/testvg-app  
mount /dev/mapper/testvg-appsnap /mnt/mirror
```

I tyle. Należy tylko pamiętać że do działa do 10G zmian na dysku, oczywiście rozmiar można odpowiedni zwiększyć/

Usuwanie:

```
umount /dev/mapper/testvg-app  
lvremove /dev/mapper/testvg-app
```

I teraz uwaga. LV z podłączonym snapshotem **będzie** miał obniżoną wydajność podczas zapisu.

Dlaczego ? Ponieważ gdy app chce zapisać np. blok nr. 2697 to LVM musi:

- 1.Odczytać ten blok z oryginalnego LV
- 2.Zapisać go w mirrorze (żeby mieć jego starą wersję w snapshotcie)
- 3.Zapisać go w oryginale.

Gdy app po raz drugi będzie chciała blok 2697 to LVM już nie będzie tego robić ponieważ ma już starą wersję na snapshotcie, ale przy pierwszym zapisie jest dość duży overhead

3.2 Mirroring

Czyli jak zrobić prosty RAID 1

przy tworzeniu LV przy pomocy lvcreate, dodać

```
-m 1
```

przy czym "1" to liczba dodatkowych kopii czyli np "-m 3" zapisze w sumie w 4 miejscach (oryginał + 4 kopie)

Oczywiście musimy mieć odpowiednią ilość PV (dysk./part. fizycznych) w VG. UWAGA! LVM nie rozróżnia czy PV dodane do VG są na tym samym dysku czy różnym więc raczej należy się trzymać zasady "1 dysk – 1 PV"

3.3 Striping (czyli mini RAID0)

Nie, nie striptiz, striping polega na rozdzieleniu jednego LV na parę dysków fizycznych, taki prosty RAIDzik0.

Dodaj przy tworzeniu LV do lvcreate:

```
-i 2
```

I raid0 gotowy. numer określa (oczywiście) na ile należy podzielić.

Dodaj:

```
-I 128
```

?eby ustalić wielkość bloku (tylko potęgi dwójki).

4.Auto-resize

No dobra wypatrywanie kiedy się skończy miejsce na dysku i potem klepanie komend z konsoli nie jest zbyt ciekawym sposobem spędzania czasu ;) Napisałem malutki program w perlu robiący to za nas:

<http://devrandom.pl/wp-content/uploads/2008/12/lvautoresize.tar.gz>

kopiujemy plik lvautoresize.conf do /etc/ i odpalamy program, wczyta on dane z configa, popatrzy na systemy plików i powiększy co trzeba. Za wszelkie spsucia przez ten program nie odpowiadam ;) Plik konfiguracyjny jest w XMLu, wszystkie dane w MBajtach. Proponuje dodać ten programik do crona, co godzinę;]. Potrafi też pisać to sysloga (jako "lvautosize"), podstawowe info, co zwiększył i czy zakończyło się to sukcesem

source: <http://devrandom.pl/blog/2008/12/27/lvm/>

Posted - Sun, May 20, 2018 7:28 PM. This article has been viewed 4645 times.

Online URL: <http://kb.ictbanking.net/article.php?id=7>