



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'lvconvert.8' command

\$ man lvconvert.8

LVCONVERT(8) System Manager's Manual LVCONVERT(8)

NAME

lvconvert ? Change logical volume layout

SYNOPSIS

lvconvert option_args position_args

[option_args]

[position_args]

--alloc contiguous|cling|cling_by_tags|normal|anywhere|inherit

-b|--background

-H|--cache

--cachedevice PV

--cachemetadatatype auto|1|2

--cachemode writethrough|writeback|passthrough

--cachepolicy String

--cachepool LV

--cachesettings String

--cachesize Size[m|UNIT]

--cachevol LV

-c|--chunksize Size[k|UNIT]

--commandprofile String

--compression y|n

--config String

-d|--debug

--deduplication y|n
--devices PV
--devicesfile String
--discards passdown|nopassdown|ignore
--driverloaded y|n
--errorwhenfull y|n
-f|--force
-h|--help
-i|--interval Number
--journal String
--lockopt String
--longhelp
--merge
--mergemirrors
--mergesnapshot
--mergethin
--metadataprofile String
--mirrorlog core|disk
-m|--mirrors [+|-]Number
-n|--name String
--nohints
--nolocking
--noudevsync
--originname LV
--poolmetadata LV
--poolmetadatasize Size[m|UNIT]
--poolmetadataspare y|n
--profile String
-q|--quiet
--raidintegrity y|n
--raidintegrityblocksize Number
--raidintegritymode String
-r|--readahead auto|none|Number

-R|--regionsize Size[m|UNIT]
 --repair
 --replace PV
 -s|--snapshot
 --splitcache
 --splitmirrors Number
 --splitsnapshot
 --startpoll
 --stripes Number
 -l|--stripesize Size[k|UNIT]
 --swapmetadata
 -t|--test
 -T|--thin
 --thinpool LV
 --trackchanges
 --type linear|striped|snapshot|raid|mirror|thin|thin-pool|vdo|
 vdo-pool|cache|cache-pool|writecache
 --uncache
 --usepolicies
 --vdopool LV
 --vdosettings String
 -v|--verbose
 --version
 -V|--virtualsize Size[m|UNIT]
 -y|--yes
 -Z|--zero y|n

DESCRIPTION

lvconvert changes the LV type and includes utilities for LV data maintenance. The LV type controls data layout and redundancy. The LV type is also called the segment type or segtype.

To display the current LV type, run the command:

```
lvs -o name,segtype LV
```

In some cases, an LV is a single device mapper (dm) layer above physi?

cal devices. In other cases, hidden LVs (dm devices) are layered between the visible LV and physical devices. LVs in the middle layers are called sub LVs. A command run on a visible LV sometimes operates on a sub LV rather than the specified LV. In other cases, a sub LV must be specified directly on the command line.

Sub LVs can be displayed with the command:

```
lvs -a
```

The linear type is equivalent to the striped type when one stripe exists. In that case, the types can sometimes be used interchangeably.

In most cases, the mirror type is deprecated and the raid1 type should be used. They are both implementations of mirroring.

Striped raid types are raid0/raid0_meta, raid5 (an alias for raid5_ls), raid6 (an alias for raid6_zr) and raid10 (an alias for raid10_near).

As opposed to mirroring, raid5 and raid6 stripe data and calculate parity blocks. The parity blocks can be used for data block recovery in case devices fail. A maximum number of one device in a raid5 LV may fail, and two in case of raid6. Striped raid types typically rotate the parity and data blocks for performance reasons, thus avoiding contention on a single device. Specific arrangements of parity and data blocks (layouts) can be used to optimize I/O performance, or to convert between raid levels. See `lvraid(7)` for more information.

Layouts of raid5 rotating parity blocks can be: left-asymmetric (raid5_la), left-symmetric (raid5_ls with alias raid5), right-asymmetric (raid5_ra), right-symmetric (raid5_rs) and raid5_n, which doesn't rotate parity blocks. Layouts of raid6 are: zero-restart (raid6_zr with alias raid6), next-restart (raid6_nr), and next-continue (raid6_nc).

Layouts including _n allow for conversion between raid levels (raid5_n to raid6 or raid5_n to striped/raid0/raid0_meta). Additionally, special raid6 layouts for raid level conversions between raid5 and raid6 are: raid6_ls_6, raid6_rs_6, raid6_la_6 and raid6_ra_6. Those correspond to their raid5 counterparts (e.g. raid5_rs can be directly converted to raid6_rs_6 and vice-versa).

raid10 (an alias for raid10_near) is currently limited to one data copy

and even number of sub LVs. This is a mirror group layout, thus a single sub LV may fail per mirror group without data loss.

Striped raid types support converting the layout, their stripesize and their number of stripes.

The striped raid types combined with raid1 allow for conversion from linear ? striped/raid0/raid0_meta and vice-versa by e.g. linear ? raid1 ? raid5_n (then adding stripes) ? striped/raid0/raid0_meta.

USAGE

Convert LV to linear.

```
lvconvert --type linear LV
```

```
[ COMMON_OPTIONS ]
```

```
[ PV ... ]
```

?

Convert LV to striped.

```
lvconvert --type striped LV
```

```
[ -l|--stripesize Size[k|UNIT] ]
```

```
[ -R|--regionsize Size[m|UNIT] ]
```

```
[ -i|--interval Number ]
```

```
[ --stripes Number ]
```

```
[ COMMON_OPTIONS ]
```

```
[ PV ... ]
```

?

Convert LV to type mirror (also see type raid1),

```
lvconvert --type mirror LV
```

```
[ -m|--mirrors [+|-]Number ]
```

```
[ -l|--stripesize Size[k|UNIT] ]
```

```
[ -R|--regionsize Size[m|UNIT] ]
```

```
[ -i|--interval Number ]
```

```
[ --stripes Number ]
```

```
[ --mirrorlog core|disk ]
```

```
[ COMMON_OPTIONS ]
```

```
[ PV ... ]
```

?

Convert LV to raid or change raid layout

(a specific raid level must be used, e.g. raid1).

`lvconvert --type raid LV`

[`-m|--mirrors` [+|-]Number]

[`-l|--stripesize` Size[k|UNIT]]

[`-R|--regionsize` Size[m|UNIT]]

[`-i|--interval` Number]

[`--stripes` Number]

[COMMON_OPTIONS]

[PV ...]

?

Convert LV to raid1 or mirror, or change number of mirror images.

`lvconvert -m|--mirrors` [+|-]Number LV

[`-R|--regionsize` Size[m|UNIT]]

[`-i|--interval` Number]

[`--mirrorlog` core|disk]

[COMMON_OPTIONS]

[PV ...]

?

Convert raid LV to change number of stripe images.

`lvconvert --stripes` Number LV1

[`-i|--interval` Number]

[`-R|--regionsize` Size[m|UNIT]]

[`-l|--stripesize` Size[k|UNIT]]

[COMMON_OPTIONS]

[PV ...]

LV1 types: raid

?

Convert raid LV to change the stripe size.

`lvconvert -l|--stripesize` Size[k|UNIT] LV1

[`-i|--interval` Number]

[`-R|--regionsize` Size[m|UNIT]]

[COMMON_OPTIONS]

LV1 types: raid

?

Split images from a raid1 or mirror LV and use them to create a new LV.

`lvconvert --splitmirrors Number -n|--name LV_new LV1`

[COMMON_OPTIONS]

[PV ...]

LV1 types: cache mirror raid1

?

Split images from a raid1 LV and track changes to origin for later merge.

`lvconvert --splitmirrors Number --trackchanges LV1`

[COMMON_OPTIONS]

[PV ...]

LV1 types: cache raid1

?

Merge LV images that were split from a raid1 LV.

`lvconvert --mergemirrors VG|LV1|Tag ...`

[COMMON_OPTIONS]

LV1 types: linear raid

?

Convert LV to a thin LV, using the original LV as an external origin.

`lvconvert --type thin --thinpool LV LV1`

[-T|--thin]

[-r|--readahead auto|none|Number]

[-c|--chunksize Size[k|UNIT]]

[-Z|--zero y|n]

[--originname LV_new]

[--poolmetadata LV]

[--poolmetadatasize Size[m|UNIT]]

[--poolmetadataspare y|n]

[--metadataprofile String]

[COMMON_OPTIONS]

LV1 types: linear striped thin cache raid error zero

?

Attach a cache pool to an LV, converts the LV to type cache.

`lvconvert --type cache --cachepool LV LV1`

- [`-H|--cache`]
- [`-Z|--zero y|n`]
- [`-r|--readahead auto|none|Number`]
- [`-c|--chunksize Size[k|UNIT]`]
- [`--cachemetadadataformat auto|1|2`]
- [`--cachemode writethrough|writeback|passthrough`]
- [`--cachepolicy String`]
- [`--cachesettings String`]
- [`--poolmetadata LV`]
- [`--poolmetadatasize Size[m|UNIT]`]
- [`--poolmetadataspare y|n`]
- [`--metadataprofile String`]
- [`COMMON_OPTIONS`]

LV1 types: linear striped thinpool vdo vdopool vdopooldata raid

?

Attach a writecache to an LV, converts the LV to type writecache.

`lvconvert --type writecache --cachevol LV LV1`

- [`--cachesettings String`]
- [`COMMON_OPTIONS`]

LV1 types: linear striped thinpool raid

?

Attach a cache to an LV, converts the LV to type cache.

`lvconvert --type cache --cachevol LV LV1`

- [`-H|--cache`]
- [`-Z|--zero y|n`]
- [`-c|--chunksize Size[k|UNIT]`]
- [`--cachemetadadataformat auto|1|2`]
- [`--cachemode writethrough|writeback|passthrough`]
- [`--cachepolicy String`]
- [`--cachesettings String`]

[--poolmetadatasize Size[m|UNIT]]

[COMMON_OPTIONS]

LV1 types: linear striped thinpool raid

?

Add a writecache to an LV, using a specified cache device.

lvconvert --type writecache --cachedevice PV LV1

[--cachesize Size[m|UNIT]]

[--cachesettings String]

[COMMON_OPTIONS]

LV1 types: linear striped thinpool raid

?

Add a cache to an LV, using a specified cache device.

lvconvert --type cache --cachedevice PV LV1

[-c|--chunksize Size[k|UNIT]]

[--cachesize Size[m|UNIT]]

[--cachesettings String]

[COMMON_OPTIONS]

LV1 types: linear striped thinpool raid

?

Convert LV to type thin-pool.

lvconvert --type thin-pool LV1

[-l|--stripesize Size[k|UNIT]]

[-r|--readahead auto|none|Number]

[-c|--chunksize Size[k|UNIT]]

[-Z|--zero y|n]

[--stripes Number]

[--discards passdown|nopassdown|ignore]

[--errorwhenfull y|n]

[--poolmetadata LV]

[--poolmetadatasize Size[m|UNIT]]

[--poolmetadataspare y|n]

[--metadataprofile String]

[COMMON_OPTIONS]

[PV ...]

LV1 types: linear striped cache raid error zero writecache

?

Convert LV to type cache-pool.

lvconvert --type cache-pool LV1

[-Z|--zero y|n]

[-r|--readahead auto|none|Number]

[-c|--chunksize Size[k|UNIT]]

[--cachemetadadataformat auto|1|2]

[--cachemode writethrough|writeback|passthrough]

[--cachepolicy String]

[--cachesettings String]

[--poolmetadata LV]

[--poolmetadatasize Size[m|UNIT]]

[--poolmetadataspare y|n]

[--metadataprofile String]

[COMMON_OPTIONS]

[PV ...]

LV1 types: linear striped raid

?

Convert LV to type vdopool.

lvconvert --type vdo-pool LV1

[-n|--name LV_new]

[-V|--virtualsize Size[m|UNIT]]

[-r|--readahead auto|none|Number]

[-Z|--zero y|n]

[--metadataprofile String]

[--compression y|n]

[--deduplication y|n]

[--vdosettings String]

[COMMON_OPTIONS]

LV1 types: linear striped cache raid

?

Detach a cache from an LV.

```
lvconvert --splitcache LV1
```

```
[ --cachesettings String ]
```

```
[ COMMON_OPTIONS ]
```

LV1 types: thinpool cache cachepool vdopool writecache

?

Merge thin LV into its origin LV.

```
lvconvert --mergethin LV1 ...
```

```
[ COMMON_OPTIONS ]
```

LV1 types: thin

?

Merge COW snapshot LV into its origin.

```
lvconvert --mergesnapshot LV1 ...
```

```
[ -i|--interval Number ]
```

```
[ COMMON_OPTIONS ]
```

LV1 types: snapshot

?

Combine a former COW snapshot (second arg) with a former origin LV (first arg) to reverse a splitsnapshot command.

```
lvconvert --type snapshot LV LV1
```

```
[ -s|--snapshot ]
```

```
[ -c|--chunksize Size[k|UNIT] ]
```

```
[ -Z|--zero y|n ]
```

```
[ COMMON_OPTIONS ]
```

LV1 types: linear striped

?

Replace failed PVs in a raid or mirror LV.

Repair a thin pool.

Repair a cache pool.

```
lvconvert --repair LV1
```

```
[ -i|--interval Number ]
```

```
[ --usepolicies ]
```

```
[ --poolmetadataspare y|n ]
```

[COMMON_OPTIONS]

[PV ...]

LV1 types: thinpool cache cachepool mirror raid

?

Replace specific PV(s) in a raid LV with another PV.

lvconvert --replace PV LV1

[COMMON_OPTIONS]

[PV ...]

LV1 types: raid

?

Poll LV to continue conversion.

lvconvert --startpoll LV1

[COMMON_OPTIONS]

LV1 types: mirror raid

?

Add or remove data integrity checksums to raid images.

lvconvert --raidintegrity y|n LV1

[--raidintegritymode String]

[--raidintegrityblocksize Number]

[COMMON_OPTIONS]

[PV ...]

LV1 types: raid

?

Common options for command:

[-b|--background]

[-f|--force]

[--alloc contiguous|cling|cling_by_tags|normal|anywhere|inherit

]

[--noudevsync]

Common options for lvm:

[-d|--debug]

[-h|--help]

[-q|--quiet]

[-t|--test]

[-v|--verbose]

[-y|--yes]

[--commandprofile String]

[--config String]

[--devices PV]

[--devicesfile String]

[--driverloaded y|n]

[--journal String]

[--lockopt String]

[--longhelp]

[--nohints]

[--nolocking]

[--profile String]

[--version]

OPTIONS

--alloc contiguous|cling|cling_by_tags|normal|anywhere|inherit

Determines the allocation policy when a command needs to allocate Physical Extents (PEs) from the VG. Each VG and LV has an allocation policy which can be changed with `vgchange/lvchange`, or overridden on the command line. `normal` applies common sense rules such as not placing parallel stripes on the same PV. `inherit` applies the VG policy to an LV. `contiguous` requires new PEs be placed adjacent to existing PEs. `cling` places new PEs on the same PV as existing PEs in the same stripe of the LV. If there are sufficient PEs for an allocation, but `normal` does not use them, `anywhere` will use them even if it reduces performance, e.g. by placing two stripes on the same PV. Optional positional PV args on the command line can also be used to limit which PVs the command will use for allocation. See `lvm(8)` for more information about allocation.

-b|--background

If the operation requires polling, this option causes the com?

mand to return before the operation is complete, and polling is done in the background.

-H|--cache

Specifies the command is handling a cache LV or cache pool. See `--type cache` and `--type cache-pool`. See `lvmcache(7)` for more information about LVM caching.

--cachedevice PV

The name of a device to use for a cache.

--cachemetadadataformat auto|1|2

Specifies the cache metadata format used by cache target.

--cachemode writethrough|writeback|passthrough

Specifies when writes to a cache LV should be considered complete. `writeback` considers a write complete as soon as it is stored in the cache pool. `writethrough` considers a write complete only when it has been stored in both the cache pool and on the origin LV. While `writethrough` may be slower for writes, it is more resilient if something should happen to a device associated with the cache pool LV. With `passthrough`, all reads are served from the origin LV (all reads miss the cache) and all writes are forwarded to the origin LV; additionally, write hits cause cache block invalidates. See `lvmcache(7)` for more information.

--cachepolicy String

Specifies the cache policy for a cache LV. See `lvmcache(7)` for more information.

--cachepool LV

The name of a cache pool.

--cachesettings String

Specifies tunable kernel options for `dm-cache` or `dm-writecache` LVs. Use the form `'option=value'` or `'option1=value option2=value'`, or repeat `--cachesettings` for each option being set. These settings override the default kernel behaviors which are usually adequate. To remove `cachesettings` and revert to the default ker?

nel behaviors, use `--cachesettings 'default'` for `dm-cache` or an empty string `--cachesettings ''` for `dm-writecache`. See `lvm?cache(7)` for more information.

`--cachesize` Size[m|UNIT]

The size of cache to use.

`--cachevol` LV

The name of a cache volume.

`-c|--chunksizesize` Size[k|UNIT]

The size of chunks in a snapshot, cache pool or thin pool. For snapshots, the value must be a power of 2 between 4 KiB and 512 KiB and the default value is 4. For a cache pool the value must be between 32 KiB and 1 GiB and the default value is 64. For a thin pool the value must be between 64 KiB and 1 GiB and the default value starts with 64 and scales up to fit the pool metadata size within 128 MiB, if the pool metadata size is not specified. The value must be a multiple of 64 KiB. See `lvmthin(7)` and `lvmcache(7)` for more information.

`--commandprofile` String

The command profile to use for command configuration. See `lvm.conf(5)` for more information about profiles.

`--compression` y|n

Controls whether compression is enabled or disabled for VDO volume. See `lvmvdo(7)` for more information about VDO usage.

`--config` String

Config settings for the command. These override `lvm.conf(5)` settings. The String arg uses the same format as `lvm.conf(5)`, or may use section/field syntax. See `lvm.conf(5)` for more information about config.

`-d|--debug` ...

Set debug level. Repeat from 1 to 6 times to increase the detail of messages sent to the log file and/or syslog (if configured).

`--deduplication` y|n

Controls whether deduplication is enabled or disabled for VDO

volume. See `lvmvdo(7)` for more information about VDO usage.

`--devices PV`

Restricts the devices that are visible and accessible to the command. Devices not listed will appear to be missing. This option can be repeated, or accepts a comma separated list of devices. This overrides the devices file.

`--devicesfile String`

A file listing devices that LVM should use. The file must exist in `/etc/lvm/devices/` and is managed with the `lvmdevices(8)` command. This overrides the `lvm.conf(5)` `devices/devicesfile` and `devices/use_devicesfile` settings.

`--discards passdown|nopassdown|ignore`

Specifies how the device-mapper thin pool layer in the kernel should handle discards. `ignore` causes the thin pool to ignore discards. `nopassdown` causes the thin pool to process discards itself to allow reuse of unneeded extents in the thin pool. `passdown` causes the thin pool to process discards itself (like `nopassdown`) and pass the discards to the underlying device. See `lvmthin(7)` for more information.

`--driverloaded y|n`

If set to `no`, the command will not attempt to use device-mapper. For testing and debugging.

`--errorwhenfull y|n`

Specifies thin pool behavior when data space is exhausted. When `yes`, device-mapper will immediately return an error when a thin pool is full and an I/O request requires space. When `no`, device-mapper will queue these I/O requests for a period of time to allow the thin pool to be extended. Errors are returned if no space is available after the timeout. (Also see `dm-thin-pool` kernel module option `no_space_timeout`.) See `lvmthin(7)` for more information.

`-f|--force ...`

Override various checks, confirmations and protections. Use

with extreme caution.

`-h|--help`

Display help text.

`-i|--interval Number`

Report progress at regular intervals.

`--journal String`

Record information in the systemd journal. This information is in addition to information enabled by the `lvm.conf log/journal` setting. `command`: record information about the command. `out?`: record the default command output. `debug`: record full command debugging.

`--lockopt String`

Used to pass options for special cases to `lvmlockd`. See `lvmlockd(8)` for more information.

`--longhelp`

Display long help text.

`--merge`

An alias for `--mergethin`, `--mergemirrors`, or `--mergesnapshot`, depending on the type of LV.

`--mergemirrors`

Merge LV images that were split from a `raid1` LV. See `--split?mirrors` with `--trackchanges`.

`--mergesnapshot`

Merge COW snapshot LV into its origin. When merging a snapshot, if both the origin and snapshot LVs are not open, the merge will start immediately. Otherwise, the merge will start the first time either the origin or snapshot LV are activated and both are closed. Merging a snapshot into an origin that cannot be closed, for example a root filesystem, is deferred until the next time the origin volume is activated. When merging starts, the resulting LV will have the origin's name, minor number and UUID. While the merge is in progress, reads or writes to the origin appear as being directed to the snapshot being merged. When the merge

finishes, the merged snapshot is removed. Multiple snapshots may be specified on the command line or a @tag may be used to specify multiple snapshots be merged to their respective origin.

--mergethin

Merge thin LV into its origin LV. The origin thin LV takes the content of the thin snapshot, and the thin snapshot LV is removed. See `lvmthin(7)` for more information.

--metadataprofile String

The metadata profile to use for command configuration. See `lvm.conf(5)` for more information about profiles.

--mirrorlog core|disk

Specifies the type of mirror log for LVs with the "mirror" type (does not apply to the "raid1" type.) disk is a persistent log and requires a small amount of storage space, usually on a separate device from the data being mirrored. core is not persistent; the log is kept only in memory. In this case, the mirror must be synchronized (by copying LV data from the first device to others) each time the LV is activated, e.g. after reboot. mirrored is a persistent log that is itself mirrored, but should be avoided. Instead, use the raid1 type for log redundancy.

-m|--mirrors [+|-]Number

Specifies the number of mirror images in addition to the original LV image, e.g. `--mirrors 1` means there are two images of the data, the original and one mirror image. Optional positional PV args on the command line can specify the devices the images should be placed on. There are two mirroring implementations: "raid1" and "mirror". These are the names of the corresponding LV types, or "segment types". Use the `--type` option to specify which to use (raid1 is default, and mirror is legacy) Use `lvm.conf(5)` `global/mirror_segtype_default` and `global/raid10_segtype_default` to configure the default types. The plus prefix + can be used, in which case the number is added to the current number of images, or the minus prefix - can be used, in which

case the number is subtracted from the current number of images.

See `lvraid(7)` for more information.

`-n|--name String`

Specifies the name of a new LV. When unspecified, a default name of "lvol#" is generated, where # is a number generated by LVM.

`--nohints`

Do not use the hints file to locate devices for PVs. A command may read more devices to find PVs when hints are not used. The command will still perform standard hint file invalidation where appropriate.

`--nolocking`

Disable locking. Use with caution, concurrent commands may produce incorrect results.

`--noudevsync`

Disables udev synchronisation. The process will not wait for notification from udev. It will continue irrespective of any possible udev processing in the background. Only use this if udev is not running or has rules that ignore the devices LVM creates.

`--originname LV`

Specifies the name to use for the external origin LV when converting an LV to a thin LV. The LV being converted becomes a read-only external origin with this name.

`--poolmetadata LV`

The name of an LV to use for storing pool metadata.

`--poolmetadatasize Size[m|UNIT]`

Specifies the size of the new pool metadata LV.

`--poolmetadataspare y|n`

Enable or disable the automatic creation and management of a spare pool metadata LV in the VG. A spare metadata LV is reserved space that can be used when repairing a pool.

`--profile String`

An alias for `--commandprofile` or `--metadataprofile`, depending on

the command.

`-q|--quiet ...`

Suppress output and log messages. Overrides `--debug` and `--verbose`. Repeat once to also suppress any prompts with answer 'no'.

`--raidintegrity y|n`

Enable or disable data integrity checksums for raid images.

`--raidintegrityblocksize Number`

The block size to use for dm-integrity on raid images. The integrity block size should usually match the device logical block size, or the file system block size. It may be less than the file system block size, but not less than the device logical block size. Possible values: 512, 1024, 2048, 4096.

`--raidintegritymode String`

Use a journal (default) or bitmap for keeping integrity checksums consistent in case of a crash. The bitmap areas are recalculated after a crash, so corruption in those areas would not be detected. A journal does not have this problem. The journal mode doubles writes to storage, but can improve performance for scattered writes packed into a single journal write. bitmap mode can in theory achieve full write throughput of the device, but would not benefit from the potential scattered write optimization.

`-r|--readahead auto|none|Number`

Sets read ahead sector count of an LV. auto is the default which allows the kernel to choose a suitable value automatically. none is equivalent to zero.

`-R|--regionsize Size[m|UNIT]`

Size of each raid or mirror synchronization region. `lvm.conf(5)` `activation/raid_region_size` can be used to configure a default.

`--repair`

Replace failed PVs in a raid or mirror LV, or run a repair utility on a thin pool. See `lvraid(7)` and `lvthin(7)` for more info.

formation.

--replace PV

Replace a specific PV in a raid LV with another PV. The new PV to use can be optionally specified after the LV. Multiple PVs can be replaced by repeating this option. See `lvraid(7)` for more information.

-s|--snapshot

Combine a former COW snapshot LV with a former origin LV to reverse a previous `--splitsnapshot` command.

--splitcache

Separates a cache pool from a cache LV, and keeps the unused cache pool LV. Before the separation, the cache is flushed. Also see `--uncache`.

--splitmirrors Number

Splits the specified number of images from a raid1 or mirror LV and uses them to create a new LV. If `--trackchanges` is also specified, changes to the raid1 LV are tracked while the split LV remains detached. If `--name` is specified, then the images are permanently split from the original LV and changes are not tracked.

--splitsnapshot

Separates a COW snapshot from its origin LV. The LV that is split off contains the chunks that differ from the origin LV along with metadata describing them. This LV can be wiped and then destroyed with `lvremove`.

--startpoll

Start polling an LV to continue processing a conversion.

--stripes Number

Specifies the number of stripes in a striped LV. This is the number of PVs (devices) that a striped LV is spread across. Data that appears sequential in the LV is spread across multiple devices in units of the stripe size (see `--stripesize`). This does not apply to existing allocated space, only newly allocated

space can be striped.

`-l|--stripesize Size[k|UNIT]`

The amount of data that is written to one device before moving to the next in a striped LV.

`--swapmetadata`

Extracts the metadata LV from a pool and replaces it with another specified LV. The extracted LV is preserved and given the name of the LV that replaced it. Use for repair only. When the metadata LV is swapped out of the pool, it can be activated directly and used with thin provisioning tools: `cache_dump(8)`, `cache_repair(8)`, `cache_restore(8)`, `thin_dump(8)`, `thin_repair(8)`, `thin_restore(8)`.

`-t|--test`

Run in test mode. Commands will not update metadata. This is implemented by disabling all metadata writing but nevertheless returning success to the calling function. This may lead to unusual error messages in multi-stage operations if a tool relies on reading back metadata it believes has changed but hasn't.

`-T|--thin`

Specifies the command is handling a thin LV or thin pool. See `--type thin`, `--type thin-pool`, and `--virtualsize`. See `lvmthin(7)` for more information about LVM thin provisioning.

`--thinpool LV`

The name of a thin pool LV.

`--trackchanges`

Can be used with `--splitmirrors` on a `raid1` LV. This causes changes to the original `raid1` LV to be tracked while the split images remain detached. This is a temporary state that allows the read-only detached image to be merged efficiently back into the `raid1` LV later. Only the regions with changed data are resynchronized during merge. While a `raid1` LV is tracking changes, operations on it are limited to merging the split image (see `--mergemirrors`) or permanently splitting the image (see

--splitmirrors with --name.

--type linear|striped|snapshot|raid|mirror|thin|thin-pool|vdo|vdo-pool|
cache|cache-pool|writecache

The LV type, also known as "segment type" or "segtype". See usage descriptions for the specific ways to use these types. For more information about redundancy and performance (raid<N>, mirror, striped, linear) see [lvmraid\(7\)](#). For thin provisioning (thin, thin-pool) see [lvmthin\(7\)](#). For performance caching (cache, cache-pool) see [lvmcache\(7\)](#). For copy-on-write snapshots (snapshot) see usage definitions. For VDO (vdo) see [lvmvdo\(7\)](#). Several commands omit an explicit type option because the type is inferred from other options or shortcuts (e.g. --stripes, --mirrors, --snapshot, --virtualsize, --thin, --cache, --vdo). Use inferred types with care because it can lead to unexpected results.

--uncache

Separates a cache pool from a cache LV, and deletes the unused cache pool LV. Before the separation, the cache is flushed. Also see --splitcache.

--usepolicies

Perform an operation according to the policy configured in [lvm.conf\(5\)](#) or a profile.

--vdopool LV

The name of a VDO pool LV. See [lvmvdo\(7\)](#) for more information about VDO usage.

--vdosettings String

Specifies tunable VDO options for VDO LVs. Use the form 'option=value' or 'option1=value option2=value', or repeat --vdosettings for each option being set. These settings override the default VDO behaviors. To remove vdosettings and revert to the default VDO behaviors, use --vdosettings 'default'. See [lvmvdo\(7\)](#) for more information.

-v|--verbose ...

Set verbose level. Repeat from 1 to 4 times to increase the default tail of messages sent to stdout and stderr.

`--version`

Display version information.

`-V|--virtualsize Size[m|UNIT]`

The virtual size of a new thin LV. See `lvmthin(7)` for more information about LVM thin provisioning. Using virtual size (`-V`) and actual size (`-L`) together creates a sparse LV. `lvm.conf(5)` `global/sparse_segtype_default` determines the default segment type used to create a sparse LV. Anything written to a sparse LV will be returned when reading from it. Reading from other areas of the LV will return blocks of zeros. When using a snapshot to create a sparse LV, a hidden virtual device is created using the zero target, and the LV has the suffix `_vorigin`. Snapshots are less efficient than thin provisioning when creating large sparse LVs (GiB).

`-y|--yes`

Do not prompt for confirmation interactively but always assume the answer yes. Use with extreme caution. (For automatic no, see `-qq`.)

`-Z|--zero y|n`

For snapshots, this controls zeroing of the first 4 KiB of data in the snapshot. If the LV is read-only, the snapshot will not be zeroed. For thin pools, this controls zeroing of provisioned blocks. Provisioning of large zeroed chunks negatively impacts performance.

VARIABLES

VG Volume Group name. See `lvm(8)` for valid names.

LV Logical Volume name. See `lvm(8)` for valid names. An LV positional arg generally includes the VG name and LV name, e.g. `VG/LV`. `LV1` indicates the LV must have a specific type, where the accepted LV types are listed. (`raid` represents `raid<N>` type).

PV Physical Volume name, a device path under /dev. For commands managing physical extents, a PV positional arg generally accepts a suffix indicating a range (or multiple ranges) of physical extents (PEs). When the first PE is omitted, it defaults to the start of the device, and when the last PE is omitted it defaults to end. Start and end range (inclusive): PV[:PE-PE]... Start and length range (counting from 0): PV[:PE+PE]...

Tag Tag name. See lvm(8) for information about tag names and using tags in place of a VG, LV or PV.

String See the option description for information about the string content.

Size[UNIT]

Size is an input number that accepts an optional unit. Input units are always treated as base two values, regardless of capitalization, e.g. 'k' and 'K' both refer to 1024. The default input unit is specified by letter, followed by |UNIT. UNIT represents other possible input units: b|B is bytes, s|S is sectors of 512 bytes, k|K is KiB, m|M is MiB, g|G is GiB, t|T is TiB, p|P is PiB, e|E is EiB. (This should not be confused with the output control --units, where capital letters mean multiple of 1000.)

ENVIRONMENT VARIABLES

See lvm(8) for information about environment variables used by lvm.

For example, LVM_VG_NAME can generally be substituted for a required VG parameter.

ADVANCED USAGE

Alternate command forms, advanced command usage, and listing of all valid syntax for completeness.

Change the region size of an LV.

lvconvert -R|--regionsize Size[m|UNIT] LV1

[COMMON_OPTIONS]

LV1 types: raid

?

Change the type of mirror log used by a mirror LV.

```
lvconvert --mirrorlog core|disk LV1
```

[COMMON_OPTIONS]

[PV ...]

LV1 types: mirror

?

Convert LV to a thin LV, using the original LV as an external origin.

```
lvconvert -T|--thin --thinpool LV LV1
```

[--type thin] (implied)

[-r|--readahead auto|none|Number]

[-c|--chunksize Size[k|UNIT]]

[-Z|--zero y|n]

[--originname LV_new]

[--poolmetadata LV]

[--poolmetadatasize Size[m|UNIT]]

[--poolmetadataspare y|n]

[--metadataprofile String]

[COMMON_OPTIONS]

LV1 types: linear striped thin cache raid error zero

?

Attach a cache pool to an LV.

```
lvconvert -H|--cache --cachepool LV LV1
```

[--type cache] (implied)

[-Z|--zero y|n]

[-r|--readahead auto|none|Number]

[-c|--chunksize Size[k|UNIT]]

[--cachemetadadataformat auto|1|2]

[--cachemode writethrough|writeback|passthrough]

[--cachepolicy String]

[--cachesettings String]

[--poolmetadata LV]

[--poolmetadatasize Size[m|UNIT]]

[--poolmetadataspare y|n]

[--metadataprofile String]

[COMMON_OPTIONS]

LV1 types: linear striped thinpool vdo vdopool vdopooldata raid

?

Attach a cache to an LV, converts the LV to type cache.

lvconvert -H|--cache --cachevol LV LV1

[-Z|--zero y|n]

[-c|--chunksize Size[k|UNIT]]

[--cachemetadadataformat auto|1|2]

[--cachemode writethrough|writeback|passthrough]

[--cachepolicy String]

[--cachesettings String]

[--poolmetadatasize Size[m|UNIT]]

[COMMON_OPTIONS]

LV1 types: linear striped thinpool raid

?

Convert LV to type vdopool.

lvconvert --vdopool LV

[--type vdo-pool] (implied)

[-r|--readahead auto|none|Number]

[-Z|--zero y|n]

[-n|--name LV_new]

[-V|--virtualsize Size[m|UNIT]]

[--metadataprofile String]

[--compression y|n]

[--deduplication y|n]

[--vdosettings String]

[COMMON_OPTIONS]

?

Detach and delete a cache from an LV.

lvconvert --uncache LV1

[--cachesettings String]

[COMMON_OPTIONS]

LV1 types: thinpool cache vdopool writecache

?

Swap metadata LV in a thin pool or cache pool (for repair only).

lvconvert --swapmetadata --poolmetadata LV LV1

[-c|--chunksize Size[k|UNIT]]

[COMMON_OPTIONS]

LV1 types: thinpool cachepool

?

Merge LV that was split from a mirror (variant, use --mergemirrors).

Merge thin LV into its origin LV (variant, use --mergethin).

Merge COW snapshot LV into its origin (variant, use --mergesnapshot).

lvconvert --merge VG|LV1|Tag ...

[-i|--interval Number]

[COMMON_OPTIONS]

LV1 types: linear striped snapshot thin raid

?

Separate a COW snapshot from its origin LV.

lvconvert --splitsnapshot LV1

[COMMON_OPTIONS]

LV1 types: snapshot

?

Combine a former COW snapshot (second arg) with a former origin LV (first arg) to reverse a splitsnapshot command.

lvconvert -s|--snapshot LV LV1

[--type snapshot] (implied)

[-c|--chunksize Size[k|UNIT]]

[-Z|--zero y|n]

[COMMON_OPTIONS]

LV1 types: linear striped

?

Poll LV to continue conversion (also see --startpoll)

or waits till conversion/mirror syncing is finished

lvconvert LV1

[COMMON_OPTIONS]

LV1 types: mirror raid

?

NOTES

This previous command syntax would perform two different operations:

```
lvconvert --thinpool LV1 --poolmetadata LV2
```

If LV1 was not a thin pool, the command would convert LV1 to a thin pool, optionally using a specified LV for metadata. But, if LV1 was already a thin pool, the command would swap the current metadata LV with LV2 (for repair purposes.)

In the same way, this previous command syntax would perform two different operations:

```
lvconvert --cachepool LV1 --poolmetadata LV2
```

If LV1 was not a cache pool, the command would convert LV1 to a cache pool, optionally using a specified LV for metadata. But, if LV1 was already a cache pool, the command would swap the current metadata LV with LV2 (for repair purposes.)

EXAMPLES

Convert a linear LV to a two-way mirror LV.

```
lvconvert --type mirror --mirrors 1 vg/lvol1
```

Convert a linear LV to a two-way RAID1 LV.

```
lvconvert --type raid1 --mirrors 1 vg/lvol1
```

Convert a mirror LV to use an in-memory log.

```
lvconvert --mirrorlog core vg/lvol1
```

Convert a mirror LV to use a disk log.

```
lvconvert --mirrorlog disk vg/lvol1
```

Convert a mirror or raid1 LV to a linear LV.

```
lvconvert --type linear vg/lvol1
```

Convert a mirror LV to a raid1 LV with the same number of images.

```
lvconvert --type raid1 vg/lvol1
```

Convert a linear LV to a two-way mirror LV, allocating new extents from specific PV ranges.

```
lvconvert --mirrors 1 vg/lvol1 /dev/sda:0-15 /dev/sdb:0-15
```

Convert a mirror LV to a linear LV, freeing physical extents from a specific PV.

```
lvconvert --type linear vg/lvol1 /dev/sda
```

Split one image from a mirror or raid1 LV, making it a new LV.

```
lvconvert --splitmirrors 1 --name lv_split vg/lvol1
```

Split one image from a raid1 LV, and track changes made to the raid1 LV while the split image remains detached.

```
lvconvert --splitmirrors 1 --trackchanges vg/lvol1
```

Merge an image (that was previously created with --splitmirrors and --trackchanges) back into the original raid1 LV.

```
lvconvert --mergemirrors vg/lvol1_rimage_1
```

Replace PV /dev/sdb1 with PV /dev/sdf1 in a raid1/4/5/6/10 LV.

```
lvconvert --replace /dev/sdb1 vg/lvol1 /dev/sdf1
```

Replace 3 PVs /dev/sd[b-d]1 with PVs /dev/sd[f-h]1 in a raid1 LV.

```
lvconvert --replace /dev/sdb1 --replace /dev/sdc1 --replace /dev/sdd1  
vg/lvol1 /dev/sd[fgh]1
```

Replace the maximum of 2 PVs /dev/sd[bc]1 with PVs /dev/sd[gh]1 in a raid6 LV.

```
lvconvert --replace /dev/sdb1 --replace /dev/sdc1 vg/lvol1 /dev/sd[gh]1
```

Convert an LV into a thin LV in the specified thin pool. The existing LV is used as an external read-only origin for the new thin LV.

```
lvconvert --type thin --thinpool vg/tpool1 vg/lvol1
```

Convert an LV into a thin LV in the specified thin pool. The existing LV is used as an external read-only origin for the new thin LV, and is renamed "external".

```
lvconvert --type thin --thinpool vg/tpool1  
--originname external vg/lvol1
```

Convert an LV to a cache pool LV using another specified LV for cache pool metadata.

```
lvconvert --type cache-pool --poolmetadata vg/poolmeta1 vg/lvol1
```

Convert an LV to a cache LV using the specified cache pool and chunk size.

```
lvconvert --type cache --cachepool vg/cpool1 -c 128 vg/lvol1
```

Detach and keep the cache pool from a cache LV.

```
lvconvert --splitcache vg/lvol1
```

Detach and remove the cache pool from a cache LV.

```
lvconvert --uncache vg/lvol1
```

SEE ALSO

lvm(8), lvm.conf(5), lvmconfig(8), lvmdevices(8),
pvchange(8), pvck(8), pvcreate(8), pvdisplay(8), pvmove(8),
pvremove(8), pvresize(8), pvs(8), pvscan(8),
vgcfgbackup(8), vgcfgrestore(8), vgchange(8), vgck(8), vgcreate(8),
vgconvert(8), vgdisplay(8), vgexport(8), vgextend(8), vgimport(8),
vgimportclone(8), vgimportdevices(8), vgmerge(8), vgmknodes(8),
vgreduce(8), vgremove(8), vgrename(8), vgs(8), vgscan(8), vgsplit(8),
lvcreate(8), lvchange(8), lvconvert(8), lvdisplay(8), lvextend(8),
lvreduce(8), lvremove(8), lvrename(8), lvresize(8), lvs(8), lvscan(8),
lvm-fullreport(8), lvm-lvpoll(8), blkdeactivate(8), lvmdump(8),
dmeventd(8), lvmpolld(8), lvmlockd(8), lvmlockctl(8), cmirrorrd(8),
lvmdbusd(8), fsadm(8),
lvmsystemid(7), lvmreport(7), lvmraid(7), lvmthin(7), lvmcache(7)

Red Hat, Inc. LVM TOOLS 2.03.17(2) (2022-11-10) LVCONVERT(8)