NAME

lvcreate − Create a logical volume

SYNOPSIS

lvcreate *option_args position_args*
[*option_args*]
[*position_args*]

−a|**−−activate y**|**n**|**ay**
−−addtag *Tag*
−−alloc contiguous|**cling**|**cling_by_tags**|**normal**|**anywhere**|**inherit**
−A|**−−autobackup y**|**n**
−H|**−−cache**
−−cachedevice *PV*
−−cachemetadataformat 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*
−C|**−−contiguous y**|**n**
−d|**−−debug**
−−deduplication y|**n**
−−discards passdown|**nopassdown**|**ignore**
−−driverloaded y|**n**
−−errorwhenfull y|**n**
−l|**−−extents** *Number*[PERCENT]
−h|**−−help**
−K|**−−ignoreactivationskip**
−−ignoremonitoring
−−lockopt *String*
−−longhelp
−j|**−−major** *Number*
−−[raid]maxrecoveryrate *Size*[k|UNIT]
−−metadataprofile *String*
−−minor *Number*
−−[raid]minrecoveryrate *Size*[k|UNIT]
−−mirrorlog core|**disk**
−m|**−−mirrors** *Number*
−−monitor y|**n**
−n|**−−name** *String*
−−nolocking
−−nosync
−−noudevsync
−p|**−−permission rw**|**r**
−M|**−−persistent y**|**n**
−−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]
−−reportformat basic|**json**
−k|**−−setactivationskip y**|**n**
−L|**−−size** *Size*[m|UNIT]
−s|**−−snapshot**
−i|**−−stripes** *Number*
−I|**−−stripesize** *Size*[k|UNIT]
−t|**−−test**
−T|**−−thin**
−−thinpool *LV*
−−type linear|**striped**|**snapshot**|**mirror**|**raid**|**thin**|**cache**|**vdo**|**thin−pool**|**cache−pool**|**vdo−pool**
−−vdo
−−vdopool *LV*
−v|**−−verbose**
−−version
−V|**−−virtualsize** *Size*[m|UNIT]
−W|**−−wipesignatures y**|**n**
−y|**−−yes**
−Z|**−−zero y**|**n**

DESCRIPTION

lvcreate creates a new LV in a VG. For standard LVs, this requires allocating logical extents from the VG's free physical extents. If there is not enough free space, the VG can be extended with other PVs (**vgextend**(8)), or existing LVs can be reduced or removed (**lvremove**(8), **lvreduce**(8).)

To control which PVs a new LV will use, specify one or more PVs as position args at the end of the command line. lvcreate will allocate physical extents only from the specified PVs.

lvcreate can also create snapshots of existing LVs, e.g. for backup purposes. The data in a new snapshot LV represents the content of the original LV from the time the snapshot was created.

RAID LVs can be created by specifying an LV type when creating the LV (see **lvmraid**(7)). Different RAID levels require different numbers of unique PVs be available in the VG for allocation.

Thin pools (for thin provisioning) and cache pools (for caching) are represented by special LVs with types thin−pool and cache−pool (see **lvmthin**(7) and **lvmcache**(7)). The pool LVs are not usable as standard block devices, but the LV names act as references to the pools.

Thin LVs are thinly provisioned from a thin pool, and are created with a virtual size rather than a physical size. A cache LV is the combination of a standard LV with a cache pool, used to cache active portions of the LV to improve performance.

VDO LVs are also provisioned volumes from a VDO pool, and are created with a virtual size rather than a physical size (see **lvmvdo**(7)).

Usage notes

In the usage section below, **−−size** *Size* can be replaced with **−−extents** *Number*. See descriptions in the options section.

In the usage section below, **−−name** is omitted from the required options, even though it is typically used. When the name is not specified, a new LV name is generated with the "lvol" prefix and a unique numeric suffix.

In the usage section below, when creating a pool and the name is omitted the new LV pool name is generated with the "vpool" for vdo−pools for prefix and a unique numeric suffix.

Pool name can be specified together with *VG* name i.e.: vg00/mythinpool.

USAGE

Create a linear LV.

lvcreate −L|−−size *Size*[m|UNIT] *VG*
[**−l|−−extents** *Number*[PERCENT]]
[**−−type linear**]
[COMMON_OPTIONS]
[*PV* ...]
-

Create a striped LV (infers −−type striped).

lvcreate −i|−−stripes *Number* **−L|−−size** *Size*[m|UNIT] *VG*
[**−l|−−extents** *Number*[PERCENT]]
[**−I|−−stripesize** *Size*[k|UNIT]]
[COMMON_OPTIONS]
[*PV* ...]
-

Create a raid1 or mirror LV (infers −−type raid1|mirror).

lvcreate −m|−−mirrors *Number* **−L|−−size** *Size*[m|UNIT] *VG*
[**−l|−−extents** *Number*[PERCENT]]
[**−R|−−regionsize** *Size*[m|UNIT]]
[**−−mirrorlog core|disk**]
[**−−[raid]minrecoveryrate** *Size*[k|UNIT]]
[**−−[raid]maxrecoveryrate** *Size*[k|UNIT]]
[COMMON_OPTIONS]
[*PV* ...]
-

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

lvcreate −−type raid −L|−−size *Size*[m|UNIT] *VG*
[**−l|−−extents** *Number*[PERCENT]]
[**−m|−−mirrors** *Number*]
[**−i|−−stripes** *Number*]
[**−I|−−stripesize** *Size*[k|UNIT]]
[**−R|−−regionsize** *Size*[m|UNIT]]
[**−−[raid]minrecoveryrate** *Size*[k|UNIT]]
[**−−[raid]maxrecoveryrate** *Size*[k|UNIT]]
[**−−raidintegrity y|n**]
[**−−raidintegritymode** *String*]
[**−−raidintegrityblocksize** *Number*]
[COMMON_OPTIONS]
[*PV* ...]
-

Create a raid10 LV.

lvcreate −m|−−mirrors *Number* **−i|−−stripes** *Number*
−L|−−size *Size*[m|UNIT] *VG*
[**−l|−−extents** *Number*[PERCENT]]

[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−R**|**−−regionsize** *Size*[m|UNIT]]
[**−−[raid]minrecoveryrate** *Size*[k|UNIT]]
[**−−[raid]maxrecoveryrate** *Size*[k|UNIT]]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a COW snapshot LV of an origin LV.

lvcreate −s|**−−snapshot −L**|**−−size** *Size*[m|UNIT] *LV*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−−type snapshot**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a thin pool.

lvcreate −−type thin−pool −L|**−−size** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−thinpool** *LV_new*]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a cache pool.

lvcreate −−type cache−pool −L|**−−size** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−H**|**−−cache**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[**−−cachemode writethrough**|**writeback**|**passthrough**]
[**−−cachepolicy** *String*]
[**−−cachesettings** *String*]
[**−−cachemetadataformat auto**|**1**|**2**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a thin LV in a thin pool (infers −−type thin).

lvcreate −V|**−−virtualsize** *Size*[m|UNIT] **−−thinpool** *LV_thinpool VG*
[**−T**|**−−thin**]
[**−−type thin**]

[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]

-

Create a thin LV that is a snapshot of an existing thin LV
(infers −−type thin).

lvcreate −s|**−−snapshot** *LV_thin*
[**−−type thin**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]

-

Create a thin LV that is a snapshot of an external origin LV.

lvcreate −−type thin −−thinpool *LV_thinpool LV*
[**−T**|**−−thin**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]

-

Create a LV that returns VDO when used.

lvcreate −−type vdo −L|**−−size** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−V**|**−−virtualsize** *Size*[m|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−vdo**]
[**−−vdopool** *LV_new*]
[**−−compression y**|**n**]
[**−−deduplication y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a thin LV, first creating a thin pool for it,
where the new thin pool is named by the −−thinpool arg.

lvcreate −−type thin −V|**−−virtualsize** *Size*[m|UNIT]
−L|**−−size** *Size*[m|UNIT] **−−thinpool** *LV_new*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−T**|**−−thin**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]

[*PV* ...]

-

Create a new LV, then attach the specified cachepool which converts the new LV to type cache.

lvcreate −−type cache −L|**−−size** *Size*[m|UNIT]
−−cachepool *LV_cachepool VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−H**|**−−cache**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[**−−cachemode writethrough**|**writeback**|**passthrough**]
[**−−cachepolicy** *String*]
[**−−cachesettings** *String*]
[**−−cachemetadataformat auto**|**1**|**2**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a new LV, then attach the specified cachevol which converts the new LV to type cache.

lvcreate −−type cache −L|**−−size** *Size*[m|UNIT]
−−cachevol *LV VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−cachemode writethrough**|**writeback**|**passthrough**]
[**−−cachepolicy** *String*]
[**−−cachesettings** *String*]
[**−−cachemetadataformat auto**|**1**|**2**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a new LV, then attach a cachevol created from the specified cache device, which converts the new LV to type cache.

lvcreate −−type cache −L|**−−size** *Size*[m|UNIT]
−−cachedevice *PV VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−cachemode writethrough**|**writeback**|**passthrough**]
[**−−cachepolicy** *String*]
[**−−cachesettings** *String*]
[**−−cachemetadataformat auto**|**1**|**2**]
[**−−cachesize** *Size*[m|UNIT]]
[COMMON_OPTIONS]

[*PV* ...]

-

Create a new LV, then attach the specified cachevol which converts the new LV to type writecache.

lvcreate −−type writecache −L|**−−size** *Size*[m|UNIT]
−−cachevol *LV VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−cachesettings** *String*]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a new LV, then attach a cachevol created from the specified cache device, which converts the new LV to type writecache.

lvcreate −−type writecache −L|**−−size** *Size*[m|UNIT]
−−cachedevice *PV VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−cachesize** *Size*[m|UNIT]]
[**−−cachesettings** *String*]
[COMMON_OPTIONS]
[*PV* ...]

-

Common options for command:
[**−a**|**−−activate y**|**n**|**ay**]
[**−A**|**−−autobackup y**|**n**]
[**−C**|**−−contiguous y**|**n**]
[**−K**|**−−ignoreactivationskip**]
[**−j**|**−−major** *Number*]
[**−n**|**−−name** *String*]
[**−p**|**−−permission rw**|**r**]
[**−M**|**−−persistent y**|**n**]
[**−r**|**−−readahead auto**|**none**|*Number*]
[**−k**|**−−setactivationskip y**|**n**]
[**−W**|**−−wipesignatures y**|**n**]
[**−Z**|**−−zero y**|**n**]
[**−−addtag** *Tag*]
[**−−alloc contiguous**|**cling**|**cling_by_tags**|**normal**|**anywhere**|**inherit**]
[**−−ignoremonitoring**]
[**−−metadataprofile** *String*]
[**−−minor** *Number*]
[**−−monitor y**|**n**]
[**−−nosync**]
[**−−noudevsync**]
[**−−reportformat basic**|**json**]

Common options for lvm:
[**−d**|**−−debug**]

[**−h**|**−−help**]
[**−q**|**−−quiet**]
[**−t**|**−−test**]
[**−v**|**−−verbose**]
[**−y**|**−−yes**]
[**−−commandprofile** *String*]
[**−−config** *String*]
[**−−driverloaded y**|**n**]
[**−−lockopt** *String*]
[**−−longhelp**]
[**−−nolocking**]
[**−−profile** *String*]
[**−−version**]

OPTIONS

−a|**−−activate y**|**n**|**ay**

Controls the active state of the new LV. **y** makes the LV active, or available. New LVs are made active by default. **n** makes the LV inactive, or unavailable, only when possible. In some cases, creating an LV requires it to be active. For example, COW snapshots of an active origin LV can only be created in the active state (this does not apply to thin snapshots). The −−zero option normally requires the LV to be active. If autoactivation **ay** is used, the LV is only activated if it matches an item in lvm.conf activation/auto_activation_volume_list. **ay** implies −−zero n and −−wipesignatures n. See **lvmlockd**(8) for more information about activation options for shared VGs.

−−addtag *Tag*

Adds a tag to a PV, VG or LV. This option can be repeated to add multiple tags at once. See **lvm**(8) for information about tags.

−−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 overriden 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.

−A|**−−autobackup y**|**n**

Specifies if metadata should be backed up automatically after a change. Enabling this is strongly advised! See **vgcfgbackup**(8) for more information.

−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 L VM caching.

−−cachedevice *PV*

The name of a device to use for a cache.

−−cachemetadataformat 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. **writethough** 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 values for a cache LV in "Key = Value" form. Repeat this option to specify multiple values. (The default values should usually be adequate.) The special string value **default** switches settings back to their default kernel values and removes them from the list of settings stored in LVM metadata. See **lvmcache**(7) for more information.

−−cachesize *Size*[m|UNIT]
The size of cache to use.

−−cachevol *LV*
The name of a cache volume.

−c|**−−chunksize** *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 4KiB and 512KiB and the default value is 4. For a cache pool the value must be between 32KiB and 1GiB and the default value is 64. For a thin pool the value must be between 64KiB and 1GiB and the default value starts with 64 and scales up to fit the pool metadata size within 128MiB, if the pool metadata size is not specified. The value must be a multiple of 64KiB. 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 disable for VDO volume. See**lvmvdo**(7) for more information about VDO usage.

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

−C|**−−contiguous y**|**n**
Sets or resets the contiguous allocation policy for LVs. Default is no contiguous allocation based on a next free principle. It is only possible to change a non-contiguous allocation policy to contiguous if all of the allocated physical extents in the LV are already contiguous.

−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 disable for VDO volume. See**lvmvdo**(7) for more information about VDO usage.

−−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.

−l|−−extents *Number*[PERCENT]
Specifies the size of the new LV in logical extents. The −−size and −−extents options are alternate methods of specifying size. The total number of physical extents used will be greater when redundant data is needed for RAID levels. An alternate syntax allows the size to be determined indirectly as a percentage of the size of a related VG, LV, or set of PVs. The suffix **%VG** denotes the total size of the VG, the suffix **%FREE** the remaining free space in the VG, and the suffix **%PVS** the free space in the specified PVs. For a snapshot, the size can be expressed as a percentage of the total size of the origin LV with the suffix **%ORIGIN** (**100%ORIGIN** provides space for the whole origin). When expressed as a percentage, the size defines an upper limit for the number of logical extents in the new LV. The precise number of logical extents in the new LV is not determined until the command has completed.

−h|−−help
Display help text.

−K|−−ignoreactivationskip
Ignore the "activation skip" LV flag during activation to allow LVs with the flag set to be activated.

−−ignoremonitoring
Do not interact with dmeventd unless −−monitor is specified. Do not use this if dmeventd is already monitoring a device.

−−lockopt *String*
Used to pass options for special cases to lvmlockd. See **lvmlockd**(8) for more information.

−−longhelp
Display long help text.

−j|−−major *Number*
Sets the major number of an LV block device.

−−[raid]maxrecoveryrate *Size*[k|UNIT]
Sets the maximum recovery rate for a RAID LV. The rate value is an amount of data per second for each device in the array. Setting the rate to 0 means it will be unbounded. See**lvmraid**(7) for more information.

−−metadataprofile *String*
The metadata profile to use for command configuration. See **lvm.conf**(5) for more information about profiles.

−−minor *Number*
Sets the minor number of an LV block device.

−−[raid]minrecoveryrate *Size*[k|UNIT]
Sets the minimum recovery rate for a RAID LV. The rate value is an amount of data per second for each device in the array. Setting the rate to 0 means it will be unbounded. See**lvmraid**(7) for more information.

−−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 global/mirror_segtype_default and global/raid10_segtype_default to configure the default types. See the −−nosync option for avoiding initial image synchronization. See**lvmraid**(7) for more information.

−−monitor y|**n**
Start (yes) or stop (no) monitoring an LV with dmeventd. dmeventd monitors kernel events for an LV, and performs automated maintenance for the LV in reponse to specific events. See **dmeventd**(8) 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.

−−nolocking
Disable locking.

−−nosync
Causes the creation of mirror, raid1, raid4, raid5 and raid10 to skip the initial synchronization. In case of mirror, raid1 and raid10, any data written afterwards will be mirrored, but the original contents will not be copied. In case of raid4 and raid5, no parity blocks will be written, though any data written afterwards will cause parity blocks to be stored. This is useful for skipping a potentially long and resource intensive initial sync of an empty mirror/raid1/raid4/raid5 and raid10 LV. This option is not valid for raid6, because raid6 relies on proper parity (P and Q Syndromes) being created during initial synchronization in order to reconstruct proper user date in case of device failures. raid0 and raid0_meta do not provide any data copies or parity support and thus do not support initial synchronization.

−−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.

−p|**−−permission rw**|**r**
Set access permission to read only **r** or read and write **rw**.

−M|**−−persistent y**|**n**
When yes, makes the specified minor number persistent.

−−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 activation/raid_region_size can be used to configure a default.

−−reportformat basic|json
Overrides current output format for reports which is defined globally by the report/output_format setting in lvm.conf. **basic** is the original format with columns and rows. If there is more than one report per command, each report is prefixed with the report name for identification. **json** produces report output in JSON format. See **lvmreport**(7) for more information.

−k|−−setactivationskip y|n
Persistently sets (yes) or clears (no) the "activation skip" flag on an LV. An LV with this flag set is not activated unless the -−ignoreactivationskip option is used by the activation command. This flag is set by default on new thin snapshot LVs. The flag is not applied to deactivation. The current value of the flag is indicated in the lvs lv_attr bits.

−L|−−size *Size*[m|UNIT]
Specifies the size of the new LV. The −−size and −−extents options are alternate methods of specifying size. The total number of physical extents used will be greater when redundant data is needed for RAID levels.

−s|−−snapshot
Create a snapshot. Snapshots provide a "frozen image" of an origin LV. The snapshot LV can be used, e.g. for backups, while the origin LV continues to be used. This option can create a COW (copy on write) snapshot, or a thin snapshot (in a thin pool.) Thin snapshots are created when the origin is a thin LV and the size option is NOT specified. Thin snapshots share the same blocks in the thin pool, and do not allocate new space from the VG. Thin snapshots are created with the "activation skip" flag, see −−setactivationskip. A thin snapshot of a non-thin "external origin" LV is created when a thin pool is specified. Unprovisioned blocks in the thin snapshot LV are read from the external origin LV. The external origin LV must be read-only. See**lvmthin**(7) for more information about LVM thin provisioning. COW snapshots are created when a size is specified. The size is allocated from space in the VG, and is the amount of space that can be used for saving COW blocks as writes occur to the origin or snapshot. The size chosen should depend upon the amount of writes that are expected; often 20% of the origin LV is enough. If COW space runs low, it can be extended with lvextend (shrinking is also allowed with lvreduce.) A small amount of the COW snapshot LV size is used to track COW block locations, so the full size is not available for COW data blocks. Use lvs to check how much space is used, and see −−monitor to to automatically extend the size to avoid running out of space.

−i|**−−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 change existing allocated space, but only applies to space being allocated by the command. When creating a RAID 4/5/6 LV, this number does not include the extra devices that are required for parity. The largest number depends on the RAID type (raid0: 64, raid10: 32, raid4/5: 63, raid6: 62), and when unspecified, the default depends on the RAID type (raid0: 2, raid10: 2, raid4/5: 3, raid6: 5.) To stripe a new raid LV across all PVs by default, see lvm.conf allocation/raid_stripe_all_devices.

−I|**−−stripesize** *Size*[k|UNIT]
The amount of data that is written to one device before moving to the next in a striped LV.

−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 L VM thin provisioning.

−−thinpool *LV*
The name of a thin pool LV.

−−type linear|**striped**|**snapshot**|**mirror**|**raid**|**thin**|**cache**|**vdo**|**thin−pool**|**cache−pool**|**vdo−pool**
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.

−−vdo
Specifies the command is handling VDO LV. See −−type vdo. See**lvmvdo**(7) for more information about VDO usage.

−−vdopool *LV*
The name of a VDO pool LV. See**lvmvdo**(7) for more information about VDO usage.

−v|**−−verbose** ...
Set verbose level. Repeat from 1 to 4 times to increase the detail 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 L VM thin provisioning. Using virtual size (−V) and actual size (−L) together creates a sparse LV. lvm.conf 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).

−W|**−−wipesignatures y**|**n**
Controls detection and subsequent wiping of signatures on new LVs. There is a prompt for each

signature detected to confirm its wiping (unless −−yes is used to override confirmations.) When not specified, signatures are wiped whenever zeroing is done (see −−zero). This behaviour can be configured with lvm.conf allocation/wipe_signatures_when_zeroing_new_lvs. If blkid wiping is used (lvm.conf allocation/use_blkid_wiping) and LVM is compiled with blkid wiping support, then the blkid(8) library is used to detect the signatures (use blkid −k to list the signatures that are recognized). Otherwise, native LVM code is used to detect signatures (only MD RAID, swap and LUKS signatures are detected in this case.) The LV is not wiped if the read only flag is set.

−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**

Controls zeroing of the first 4KiB of data in the new LV. Default is **y**. Snapshot COW volumes are always zeroed. For thin pools, this controls zeroing of provisioned blocks. LV is not zeroed if the read only flag is set. Warning: trying to mount an unzeroed LV can cause the system to hang.

VARIABLES

VG

Volume Group name. See **lvm**(8) for valid names. For lvcreate, the required VG positional arg may be omitted when the VG name is included in another option, e.g. −−name VG/LV.

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. LV followed by _<type> indicates that an LV of the given type is required. (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*]...

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:**bBsSkKmMgGtTpPeE**. 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.

Create an LV that returns errors when used.

lvcreate −−type error −L|**−−size** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[COMMON_OPTIONS]

-

Create an LV that returns zeros when read.

lvcreate −−type zero −L|**−−size** *Size*[m|UNIT] *VG*

[**−l**|**−−extents** *Number*[PERCENT]]
[COMMON_OPTIONS]
-

Create a linear LV.

lvcreate −−type linear −L|**−−size** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[COMMON_OPTIONS]
[*PV* ...]
-

Create a striped LV (also see lvcreate −−stripes).

lvcreate −−type striped −L|**−−size** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[COMMON_OPTIONS]
[*PV* ...]
-

Create a mirror LV (also see −−type raid1).

lvcreate −−type mirror −L|**−−size** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−m**|**−−mirrors** *Number*]
[**−R**|**−−regionsize** *Size*[m|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−mirrorlog core**|**disk**]
[COMMON_OPTIONS]
[*PV* ...]
-

Create a COW snapshot LV of an origin LV
(also see −−snapshot).

lvcreate −−type snapshot −L|**−−size** *Size*[m|UNIT] *LV*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−s**|**−−snapshot**]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[COMMON_OPTIONS]
[*PV* ...]
-

Create a sparse COW snapshot LV of a virtual origin LV
(also see −−snapshot).

lvcreate −−type snapshot −L|**−−size** *Size*[m|UNIT]
−V|**−−virtualsize** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−s**|**−−snapshot**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[COMMON_OPTIONS]

[*PV* ...]

-

Create a sparse COW snapshot LV of a virtual origin LV.

lvcreate **−s**|**−−snapshot** **−L**|**−−size** *Size*[m|UNIT]
−V|**−−virtualsize** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−−type snapshot**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a thin pool (infers −−type thin−pool).

lvcreate **−T**|**−−thin** **−L**|**−−size** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−type thin−pool**]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare** **y**|**n**]
[**−−discards** **passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull** **y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a thin pool named by the −−thinpool arg
(infers −−type thin−pool).

lvcreate **−L**|**−−size** *Size*[m|UNIT] **−−thinpool** *LV_new VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−T**|**−−thin**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−type thin−pool**]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare** **y**|**n**]
[**−−discards** **passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull** **y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a cache pool named by the −−cachepool arg
(variant, uses −−cachepool in place of −−name).

lvcreate **−−type cache−pool** **−L**|**−−size** *Size*[m|UNIT]
−−cachepool *LV_new VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−H**|**−−cache**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−−poolmetadatasize** *Size*[m|UNIT]]

[**−−poolmetadataspare y**|**n**]
[**−−cachemode writethrough**|**writeback**|**passthrough**]
[**−−cachepolicy** *String*]
[**−−cachesettings** *String*]
[**−−cachemetadataformat auto**|**1**|**2**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a thin LV in a thin pool.

lvcreate −−type thin −V|**−−virtualsize** *Size*[m|UNIT]
−−thinpool *LV_thinpool VG*
[**−T**|**−−thin**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]

-

Create a thin LV in a thin pool named in the first arg
(variant, also see −−thinpool for naming pool).

lvcreate −−type thin −V|**−−virtualsize** *Size*[m|UNIT] *LV_thinpool*
[**−T**|**−−thin**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]

-

Create a thin LV in the thin pool named in the first arg
(variant, infers −−type thin, also see −−thinpool for
naming pool.)

lvcreate −V|**−−virtualsize** *Size*[m|UNIT] *LV_thinpool*
[**−T**|**−−thin**]
[**−−type thin**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]

-

Create a thin LV that is a snapshot of an existing thin LV.

lvcreate −−type thin *LV_thin*
[**−T**|**−−thin**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]

-

Create a thin LV that is a snapshot of an existing thin LV
(infers −−type thin).

lvcreate −T|**−−thin** *LV_thin*

[**−−type thin**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]

-

Create a thin LV that is a snapshot of an external origin LV (infers −−type thin).

lvcreate −s|**−−snapshot −−thinpool** *LV_thinpool LV*
[**−−type thin**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]

-

Create a VDO LV with VDO pool.

lvcreate −−vdo −L|**−−size** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−V**|**−−virtualsize** *Size*[m|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−vdopool** *LV_new*]
[**−−compression y**|**n**]
[**−−deduplication y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a VDO LV with VDO pool.

lvcreate −−vdopool *LV_new* **−L**|**−−size** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−V**|**−−virtualsize** *Size*[m|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−vdo**]
[**−−type vdo**]
[**−−compression y**|**n**]
[**−−deduplication y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a thin LV, first creating a thin pool for it, where the new thin pool is named by the −−thinpool arg (variant, infers −−type thin).

lvcreate −V|**−−virtualsize** *Size*[m|UNIT] **−L**|**−−size** *Size*[m|UNIT]
−−thinpool *LV_new*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−T**|**−−thin**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−poolmetadatasize** *Size*[m|UNIT]]

[**−−poolmetadataspare y**|**n**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a thin LV, first creating a thin pool for it,
where the new thin pool is named by the −−thinpool arg
(variant, infers −−type thin).

lvcreate −V|**−−virtualsize** *Size*[m|UNIT] **−L**|**−−size** *Size*[m|UNIT]
−−thinpool *LV_new VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−T**|**−−thin**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a thin LV, first creating a thin pool for it,
where the new thin pool is named in the first arg,
or the new thin pool name is generated when the first
arg is a VG name.

lvcreate −−type thin −V|**−−virtualsize** *Size*[m|UNIT]
−L|**−−size** *Size*[m|UNIT] *VG*|*LV_new*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−T**|**−−thin**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a thin LV, first creating a thin pool for it,
where the new thin pool is named in the first arg,
or the new thin pool name is generated when the first
arg is a VG name (variant, infers −−type thin).

lvcreate −T|**−−thin −V**|**−−virtualsize** *Size*[m|UNIT]
−L|**−−size** *Size*[m|UNIT] *VG*|*LV_new*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]

[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a thin LV, first creating a thin pool for it
(infers −−type thin).
Create a sparse snapshot of a virtual origin LV
(infers −−type snapshot).
Chooses −−type thin or −−type snapshot according to
config setting sparse_segtype_default.

lvcreate −L|**−−size** *Size*[m|UNIT] **−V**|**−−virtualsize** *Size*[m|UNIT] *VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−s**|**−−snapshot**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−type snapshot**]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[**−−discards passdown**|**nopassdown**|**ignore**]
[**−−errorwhenfull y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a new LV, then attach the specified cachepool
which converts the new LV to type cache
(variant, infers −−type cache.)

lvcreate −L|**−−size** *Size*[m|UNIT] **−−cachepool** *LV_cachepool VG*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−H**|**−−cache**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−type cache**]
[**−−cachemode writethrough**|**writeback**|**passthrough**]
[**−−cachepolicy** *String*]
[**−−cachesettings** *String*]
[**−−cachemetadataformat auto**|**1**|**2**]
[COMMON_OPTIONS]
[*PV* ...]

-

Create a new LV, then attach the specified cachepool
which converts the new LV to type cache.
(variant, also use −−cachepool).

lvcreate −−type cache −L|**−−size** *Size*[m|UNIT] *LV_cachepool*
[**−l**|**−−extents** *Number*[PERCENT]]

[**−H**|**−−cache**]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[**−−cachemode writethrough**|**writeback**|**passthrough**]
[**−−cachepolicy** *String*]
[**−−cachesettings** *String*]
[**−−cachemetadataformat auto**|**1**|**2**]
[COMMON_OPTIONS]
[*PV* ...]

-

When the LV arg is a cachepool, then create a new LV and attach the cachepool arg to it.
(variant, use −−type cache and −−cachepool.)
When the LV arg is not a cachepool, then create a new cachepool and attach it to the LV arg (alternative, use lvconvert.)

lvcreate −H|**−−cache −L**|**−−size** *Size*[m|UNIT] *LV*
[**−l**|**−−extents** *Number*[PERCENT]]
[**−c**|**−−chunksize** *Size*[k|UNIT]]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−cachemode writethrough**|**writeback**|**passthrough**]
[**−−cachepolicy** *String*]
[**−−cachesettings** *String*]
[**−−cachemetadataformat auto**|**1**|**2**]
[**−−poolmetadatasize** *Size*[m|UNIT]]
[**−−poolmetadataspare y**|**n**]
[COMMON_OPTIONS]
[*PV* ...]

-

EXAMPLES

Create a striped LV with 3 stripes, a stripe size of 8KiB and a size of 100MiB. The LV name is chosen by lvcreate.
lvcreate −i 3 −I 8 −L 100m vg00

Create a raid1 LV with two images, and a useable size of 500 MiB. This operation requires two devices, one for each mirror image. RAID metadata (superblock and bitmap) is also included on the two devices.
lvcreate −−type raid1 −m1 −L 500m −n mylv vg00

Create a mirror LV with two images, and a useable size of 500 MiB. This operation requires three devices: two for mirror images and one for a disk log.
lvcreate −−type mirror −m1 −L 500m −n mylv vg00

Create a mirror LV with 2 images, and a useable size of 500 MiB. This operation requires 2 devices because the log is in memory.
lvcreate −−type mirror −m1 −−mirrorlog core −L 500m −n mylv vg00

Create a copy-on−write snapshot of an LV:
lvcreate −−snapshot −−size 100m −−name mysnap vg00/mylv

Create a copy-on−write snapshot with a size sufficient for overwriting 20% of the size of the original LV.
lvcreate −s −l 20%ORIGIN −n mysnap vg00/mylv

Create a sparse LV with 1TiB of virtual space, and actual space just under 100MiB.
lvcreate −−snapshot −−virtualsize 1t −−size 100m −−name mylv vg00

Create a linear LV with a usable size of 64MiB on specific physical extents.
lvcreate −L 64m −n mylv vg00 /dev/sda:0−7 /dev/sdb:0−7

Create a RAID5 LV with a usable size of 5GiB, 3 stripes, a stripe size of 64KiB, using a total of 4 devices (including one for parity).
lvcreate −−type raid5 −L 5G −i 3 −I 64 −n mylv vg00

Create a RAID5 LV using all of the free space in the VG and spanning all the PVs in the VG (note that the command will fail if there are more than 8 PVs in the VG, in which case **−i 7** must be used to get to the current maximum of 8 devices including parity for RaidLVs).
lvcreate −−config allocation/raid_stripe_all_devices=1
−−type raid5 −l 100%FREE −n mylv vg00

Create RAID10 LV with a usable size of 5GiB, using 2 stripes, each on a two-image mirror. (Note that the **−i** and **−m** arguments behave differently: **−i** specifies the total number of stripes, but **−m** specifies the number of images in addition to the first image).
lvcreate −−type raid10 −L 5G −i 2 −m 1 −n mylv vg00

Create a 1TiB thin LV mythin, with 256GiB thinpool tpool0 in vg00.
lvcreate −−T −−size 256G −−name mythin vg00/tpool0

Create a 1TiB thin LV, first creating a new thin pool for it, where the thin pool has 100MiB of space, uses 2 stripes, has a 64KiB stripe size, and 256KiB chunk size.
lvcreate −−type thin −−name mylv −−thinpool mypool
−V 1t −L 100m −i 2 −I 64 −c 256 vg00

Create a thin snapshot of a thin LV (the size option must not be used, otherwise a copy-on−write snapshot would be created).
lvcreate −−snapshot −−name mysnap vg00/thinvol

Create a thin snapshot of the read-only inactive LV named "origin" which becomes an external origin for the thin snapshot LV.
lvcreate −−snapshot −−name mysnap −−thinpool mypool vg00/origin

Create a cache pool from a fast physical device. The cache pool can then be used to cache an LV.
lvcreate −−type cache−pool −L 1G −n my_cpool vg00 /dev/fast1

Create a cache LV, first creating a new origin LV on a slow physical device, then combining the new origin LV with an existing cache pool.
lvcreate −−type cache −−cachepool my_cpool
−L 100G −n mylv vg00 /dev/slow1

Create a VDO LV vdo0 with VDOPoolLV size of 10GiB and name vpool1.
lvcreate −−vdo −−size 10G −−name vdo0 vg00/vpool1

SEE ALSO

lvm(8) **lvm.conf**(5) **lvmconfig**(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) **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) **lvm2−activation−generator**(8) **blkdeactivate**(8) **lvmdump**(8)

dmeventd(8) **lvmpolld**(8) **lvmlockd**(8) **lvmlockctl**(8) **cmirrord**(8) **lvmdbusd**(8)

lvmsystemid(7) **lvmreport**(7) **lvmraid**(7) **lvmthin**(7) **lvmcache**(7)