NAME
lvextend – Add space to a logical volume

SYNOPSIS
lvextend *option_args position_args*
  [*option_args*]
  [*position_args*]

  −−alloc contiguous|cling|cling_by_tags|normal|anywhere|inherit
 −A|−−autobackup y|n
  −−commandprofile *String*
  −−config *String*
 −d|−−debug
  −−driverloaded y|n
 −l|−−extents [+]*Number*[PERCENT]
 −f|−−force
 −h|−−help
  −−lockopt *String*
  −−longhelp
 −m|−−mirrors *Number*
 −n|−−nofsck
  −−nolocking
  −−nosync
  −−noudevsync
  −−poolmetadatasize [+]*Size*[m|UNIT]
  −−profile *String*
 −q|−−quiet
  −−reportformat basic|json
 −r|−−resizefs
 −L|−−size [+]*Size*[m|UNIT]
 −i|−−stripes *Number*
 −I|−−stripesize *Size*[k|UNIT]
 −t|−−test
  −−type linear|striped|snapshot|mirror|raid|thin|cache|vdo|thin−pool|cache−pool|vdo−pool
  −−usepolicies
 −v|−−verbose
  −−version
 −y|−−yes

DESCRIPTION
lvextend extends the size of an LV. This requires allocating logical extents from the VG's free physical extents. If the extension adds a new LV segment, the new segment will use the existing segment type of the LV.

Extending a copy-on−write snapshot LV adds space for COW blocks.

Use **lvconvert**(8) to change the number of data images in a RAID or mirrored LV.

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

USAGE
Extend an LV by a specified size.

lvextend −L|−−size [+]*Size*[m|UNIT] *LV*
  [**−l|−−extents** [+]*Number*[PERCENT]]
  [**−r|−−resizefs**]

[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[**−−poolmetadatasize** [+]*Size*[m|UNIT]]
[COMMON_OPTIONS]
[*PV ...*]
-

Extend an LV by specified PV extents.

lvextend *LV PV* ...
[**−r**|**−−resizefs**]
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[COMMON_OPTIONS]
-

Extend a pool metadata SubLV by a specified size.

lvextend −−poolmetadatasize [+]*Size*[m|UNIT] *LV_thinpool*
[**−i**|**−−stripes** *Number*]
[**−I**|**−−stripesize** *Size*[k|UNIT]]
[COMMON_OPTIONS]
[*PV ...*]
-

Extend an LV according to a predefined policy.

lvextend −−usepolicies *LV_snapshot_thinpool*
[**−r**|**−−resizefs**]
[COMMON_OPTIONS]
[*PV ...*]
-

Common options for command:
[**−A**|**−−autobackup y**|**n**]
[**−f**|**−−force**]
[**−m**|**−−mirrors** *Number*]
[**−n**|**−−nofsck**]
[**−−alloc contiguous**|**cling**|**cling_by_tags**|**normal**|**anywhere**|**inherit**]
[**−−nosync**]
[**−−noudevsync**]
[**−−reportformat basic**|**json**]
[**−−type linear**|**striped**|**snapshot**|**mirror**|**raid**|**thin**|**cache**|**vdo**|**thin−pool**|**cache−pool**|**vdo−pool**]

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

    −−**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.

    −−**commandprofile** *String*

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

    −−**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.

    −**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).

    −−**driverloaded y|n**

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

    −**l**|−−**extents** [+]*Number*[PERCENT]

        Specifies the new size of the 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. When the plus + or minus − prefix is used, the value is not an absolute size, but is relative and added or subtracted from the current size.

    −**f**|−−**force** ...

        Override various checks, confirmations and protections. Use with extreme caution.

    −**h**|−−**help**

        Display help text.

    −−**lockopt** *String*

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

    −−**longhelp**

        Display long help text.

−m|−−mirrors *Number*
>   Not used.

−n|−−nofsck
>   Do not perform fsck before resizing filesystem when filesystem requires it. You may need to use
>   −−force to proceed with this option.

−−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 con-
>   tents 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 poten-
>   tially 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 fail-
>   ures. raid0 and raid0_meta do not provide any data copies or parity support and thus do not sup-
>   port initial synchronization.

−−noudevsync
>   Disables udev synchronisation. The process will not wait for notification from udev. It will con-
>   tinue 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.

−−poolmetadatasize [+]*Size*[m|UNIT]
>   Specifies the new size of the pool metadata LV. The plus prefix + can be used, in which case the
>   value is added to the current size.

−−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 sup-
>   press any prompts with answer 'no'.

−−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.

−r|−−resizefs
>   Resize underlying filesystem together with the LV using fsadm(8).

−L|−−size [+]*Size*[m|UNIT]
>   Specifies the new size of the LV. The −−size and −−extents options are alternate methods of speci-
>   fying size. The total number of physical extents used will be greater when redundant data is
>   needed for RAID levels. When the plus + or minus − prefix is used, the value is not an absolute
>   size, but is relative and added or subtracted from the current size.

−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 de-
>   pends 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.

−−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.

−−usepolicies
>   Perform an operation according to the policy configured in lvm.conf or a profile.

−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.

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

VARIABLES

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: **bBsSkKmMg-GtTpPeE**. 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.

EXAMPLES

Extend the size of an LV by 54MiB, using a specific PV.
lvextend −L +54 vg01/lvol10 /dev/sdk3

Extend the size of an LV by the amount of free space on PV /dev/sdk3. This is equivalent to specifying "−l +100%PVS" on the command line.
lvextend vg01/lvol01 /dev/sdk3

Extend an LV by 16MiB using specific physical extents.
lvextend −L+16m vg01/lvol01 /dev/sda:8−9 /dev/sdb:8−9

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)