NAME

pvcreate − Initialize physical volume(s) for use by LVM

SYNOPSIS

pvcreate *position_args*
  [*option_args*]

DESCRIPTION

pvcreate initializes a Physical Volume (PV) on a device so the device is recognized as belonging to LVM. This allows the PV to be used in a Volume Group (VG). An LVM disk label is written to the device, and LVM metadata areas are initialized. A PV can be placed on a whole device or partition.

Use **vgcreate**(8) to create a new VG on the PV, or **vgextend**(8) to add the PV to an existing VG. Use **pvremove**(8) to remove the LVM disk label from the device.

The force option will create a PV without confirmation. Repeating the force option (**-ff**) will forcibly create a PV, overriding checks that normally prevent it, e.g. if the PV is already in a VG.

Metadata location, size, and alignment

The LVM disk label begins 512 bytes from the start of the device, and is 512 bytes in size.

The LVM metadata area begins at an offset (from the start of the device) equal to the page size of the machine creating the PV (often 4 KiB.) The metadata area contains a 512 byte header and a multi-KiB circular buffer that holds text copies of the VG metadata.

With default settings, the first physical extent (PE), which contains LV data, is 1 MiB from the start of the device. This location is controlled by **default_data_alignment** in lvm.conf, which is set to 1 (MiB) by default. The pe_start will be a multiple of this many MiB. This location can be checked with:
pvs −o pe_start *PV*

The size of the LVM metadata area is the space between the the start of the metadata area and the first PE. When metadata begins at 4 KiB and the first PE is at 1024 KiB, the metadata area size is 1020 KiB. This can be checked with:
pvs −o mda_size *PV*

The mda_size cannot be increased after pvcreate, so if larger metadata is needed, it must be set during pvcreate. Two copies of the VG metadata must always fit within the metadata area, so the maximum VG metadata size is around half the mda_size. This can be checked with:
vgs −o mda_free *VG*

A larger metadata area can be set with −−metadatasize. The resulting mda_size may be larger than specified due to default_data_alignment placing pe_start on a MiB boundary, and the fact that the metadata area extends to the first PE. With metadata starting at 4 KiB and default_data_alignment 1 (MiB), setting −−metadatasize 2048k results in pe_start of 3 MiB and mda_size of 3068 KiB. Alternatively, −−metadatasize 2044k results in pe_start at 2 MiB and mda_size of 2044 KiB.

The alignment of pe_start described above may be automatically overriden based on md device properties or device i/o properties reported in sysfs. These automatic adjustments can be enabled/disabled using lvm.conf settings md_chunk_alignment and data_alignment_offset_detection.

To use a different pe_start alignment, use the −−dataalignment option. The −−metadatasize option would also typically be used in this case because the metadata area size also determines the location of pe_start. When using these two options together, pe_start is calculated as: metadata area start (page size), plus the specified −−metadatasize, rounded up to the next multiple of −−dataalignment. With metadata starting at 4 KiB, −−metadatasize 2048k, and −−dataalignment 128k, pe_start is 2176 KiB and mda_size is 2172 KiB.

The pe_start of 2176 KiB is the nearest even multiple of 128 KiB that provides at least 2048 KiB of metadata space. Always check the resulting alignment and metadata size when using these options.

To shift an aligned pe_start value, use the −−dataaligmentoffset option. The pe_start alignment is calculated as described above, and then the value specified with −−dataaligmentoffset is added to produce the final pe_start value.

USAGE

pvcreate *PV* ...
    [**−f**|**−−force**]
    [**−M**|**−−metadatatype lvm2**]
    [**−u**|**−−uuid** *String*]
    [**−Z**|**−−zero y**|**n**]
    [**−−dataalignment** *Size*[k|UNIT]]
    [**−−dataalignmentoffset** *Size*[k|UNIT]]
    [**−−bootloaderareasize** *Size*[m|UNIT]]
    [**−−labelsector** *Number*]
    [**−−[pv]metadatacopies 0**|**1**|**2**]
    [**−−metadatasize** *Size*[m|UNIT]]
    [**−−metadataignore y**|**n**]
    [**−−norestorefile**]
    [**−−setphysicalvolumesize** *Size*[m|UNIT]]
    [**−−reportformat basic**|**json**]
    [**−−restorefile** *String*]
    [COMMON_OPTIONS]

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

−−bootloaderareasize *Size*[m|UNIT]
: Reserve space for the bootloader between the LVM metadata area and the first PE. The bootloader area is reserved for bootloaders to embed their own data or metadata; LVM will not use it. The bootloader area begins where the first PE would otherwise be located. The first PE is moved out by the size of the bootloader area, and then moved out further if necessary to match the data alignment. The start of the bootloader area is always aligned, see also −−dataalignment and −−dataalignmentoffset. The bootloader area may be larger than requested due to the alignment, but it's never less than the requested size. To see the bootloader area start and size of an existing PV use pvs −o +pv_ba_start,pv_ba_size.

−−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.

−−dataalignment *Size*[k|UNIT]
Align the start of a PV data area with a multiple of this number. To see the location of the first Physical Extent (PE) of an existing PV, use pvs −o +pe_start. In addition, it may be shifted by an alignment offset, see −−dataalignmentoffset. Also specify an appropriate PE size when creating a VG.

−−dataalignmentoffset *Size*[k|UNIT]
Shift the start of the PV data area by this additional offset.

−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.

−f|−−force ...
Override various checks, confirmations and protections. Use with extreme caution.

−h|−−help
Display help text.

−−labelsector *Number*
By default the PV is labelled with an LVM2 identifier in its second sector (sector 1). This lets you use a different sector near the start of the disk (between 0 and 3 inclusive − see LA-BEL_SCAN_SECTORS in the source). Use with care.

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

−−longhelp
Display long help text.

−−metadataignore y|n
Specifies the metadataignore property of a PV. If yes, metadata areas on the PV are ignored, and lvm will not store metadata in the metadata areas of the PV. If no, lvm will store metadata on the PV.

−−metadatasize *Size*[m|UNIT]
The approximate amount of space used for each VG metadata area. The size may be rounded.

−M|−−metadatatype lvm2
Specifies the type of on-disk metadata to use. **lvm2** (or just **2**) is the current, standard format. **lvm1** (or just **1**) is no longer used.

−−nolocking
Disable locking.

−−norestorefile
In conjunction with −−uuid, this allows a uuid to be specified without also requiring that a backup of the metadata be provided.

−−profile *String*
An alias for −−commandprofile or −−metadataprofile, depending on the command.

−−[pv]metadatacopies 0|1|2
The number of metadata areas to set aside on a PV for storing VG metadata. When 2, one copy of the VG metadata is stored at the front of the PV and a second copy is stored at the end. When 1, one copy of the VG metadata is stored at the front of the PV. When 0, no copies of the VG

metadata are stored on the given PV. This may be useful in VGs containing many PVs (this places limitations on the ability to use vgsplit later.)

−q|−−quiet ...
: Suppress output and log messages. Overrides −−debug and −−verbose. Repeat once to also suppress 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.

−−restorefile *String*
: In conjunction with −−uuid, this reads the file (produced by vgcfgbackup), extracts the location and size of the data on the PV, and ensures that the metadata produced by the program is consistent with the contents of the file, i.e. the physical extents will be in the same place and not be overwritten by new metadata. This provides a mechanism to upgrade the metadata format or to add/remove metadata areas. Use with care.

−−setphysicalvolumesize *Size*[m|UNIT]
: Overrides the automatically detected size of the PV. Use with care, or prior to reducing the physical size of the device.

−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.

−u|−−uuid *String*
: Specify a UUID for the device. Without this option, a random UUID is generated. This option is needed before restoring a backup of LVM metadata onto a replacement device; see **vgcfgrestore**(8). As such, use of --restorefile is compulsory unless the −−norestorefile is used. All PVs must have unique UUIDs, and LVM will prevent certain operations if multiple devices are seen with the same UUID. See **vgimportclone**(8) for more information.

−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.)

−Z|−−zero y|n
: Controls if the first 4 sectors (2048 bytes) of the device are wiped. The default is to wipe these sectors unless either or both of --restorefile or −−uuid are specified.

VARIABLES

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 kilobytes, m|M is megabytes, g|G is gigabytes, t|T is terabytes, p|P is petabytes, e|E is exabytes. (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

Initialize a partition and a full device.
pvcreate /dev/sdc4 /dev/sde

If a device is a 4KiB sector drive that compensates for windows partitioning (sector 7 is the lowest aligned logical block, the 4KiB sectors start at LBA −1, and consequently sector 63 is aligned on a 4KiB boundary) manually account for this when initializing for use by LVM.
pvcreate −−dataalignmentoffset 7s /dev/sdb

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)