NAME

vgcreate − Create a volume group

SYNOPSIS

vgcreate *position_args*
[*option_args*]

DESCRIPTION

vgcreate creates a new VG on block devices. If the devices were not previously intialized as PVs with **pvcreate**(8), vgcreate will inititialize them, making them PVs. The pvcreate options for initializing devices are also available with vgcreate.

When vgcreate uses an existing PV, that PV's existing values for metadata size, PE start, etc, are used, even if different values are specified in the vgcreate command. To change these values, first use pvremove on the device.

USAGE

vgcreate *VG_new PV ...*
[**−A**|**−−autobackup y**|**n**]
[**−c**|**−−clustered y**|**n**]
[**−l**|**−−maxlogicalvolumes** *Number*]
[**−p**|**−−maxphysicalvolumes** *Number*]
[**−M**|**−−metadatatype lvm2**]
[**−s**|**−−physicalextentsize** *Size*[m|UNIT]]
[**−f**|**−−force**]
[**−Z**|**−−zero y**|**n**]
[**−−addtag** *Tag*]
[**−−alloc contiguous**|**cling**|**cling_by_tags**|**normal**|**anywhere**|**inherit**]
[**−−metadataprofile** *String*]
[**−−labelsector** *Number*]
[**−−metadatasize** *Size*[m|UNIT]]
[**−−pvmetadatacopies 0**|**1**|**2**]
[**−−[vg]metadatacopies all**|**unmanaged**|*Number*]
[**−−reportformat basic**|**json**]
[**−−dataalignment** *Size*[k|UNIT]]
[**−−dataalignmentoffset** *Size*[k|UNIT]]
[**−−shared**]
[**−−systemid** *String*]
[**−−locktype sanlock**|**dlm**|**none**]
[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

−−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.

−c|−−clustered y|n
This option was specific to clvm and is now replaced by the −−shared option with **lvmlockd**(8).

−−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 LABEL_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.

−−locktype sanlock|dlm|none
Specify the VG lock type directly in place of using −−shared. See **lvmlockd**(8) for more information.

−−longhelp
Display long help text.

−l|−−maxlogicalvolumes *Number*
Sets the maximum number of LVs allowed in a VG.

−p|−−maxphysicalvolumes *Number*
Sets the maximum number of PVs that can belong to the VG. The value 0 removes any limitation. For large numbers of PVs, also see options −−pvmetadatacopies, and −−vgmetadatacopies for improving performance.

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

−−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.

−s|−−physicalextentsize *Size*[m|UNIT]
Sets the physical extent size of PVs in the VG. The value must be either a power of 2 of at least 1 sector (where the sector size is the largest sector size of the PVs currently used in the VG), or at least 128KiB. Once this value has been set, it is difficult to change without recreating the VG, unless no extents need moving.

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

−−pvmetadatacopies 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.

−−shared
Create a shared VG using lvmlockd if LVM is compiled with lockd support. lvmlockd will select lock type sanlock or dlm depending on which lock manager is running. This allows multiple hosts to share a VG on shared devices. lvmlockd and a lock manager must be configured and running. See **lvmlockd**(8) for more information about shared VGs.

−−systemid *String*
Specifies the system ID that will be given to the new VG, overriding the system ID of the host

running the command. A VG is normally created without this option, in which case the new VG is given the system ID of the host creating it. Using this option requires caution because the system ID of the new VG may not match the system ID of the host running the command, leaving the VG inaccessible to the host. See **lvmsystemid**(7) for more information.

−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.

−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.

−−[vg]metadatacopies all|**unmanaged**|*Number*
Number of copies of the VG metadata that are kept. VG metadata is kept in VG metadata areas on PVs in the VG, i.e. reserved space at the start and/or end of the PVs. Keeping a copy of the VG metadata on every PV can reduce performance in VGs containing a large number of PVs. When this number is set to a non-zero value, LVM will automatically choose PVs on which to store metadata, using the metadataignore flags on PVs to achieve the specified number. The number can also be replaced with special string values: **unmanaged** causes LVM to not automatically manage the PV metadataignore flags. **all** causes LVM to first clear the metadataignore flags on all PVs, and then to become unmanaged.

−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

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

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.

EXAMPLES

Create a VG with two PVs, using the default physical extent size.
vgcreate myvg /dev/sdk1 /dev/sdl1

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)