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 'lvmautoactivation.7' command

$ man lvmautoactivation.7

LVMAUTOACTIVATION(7) LVMAUTOACTIVATION(7)

NAME

lvmautoactivation ? LVM autoactivation

DESCRIPTION

Autoactivation is the activation of LVs performed automatically by the system in response to LVM devices being attached to the machine. When all PVs in a VG have been attached, the VG is complete, and LVs in the VG are activated.

Autoactivation of VGs, or specific LVs, can be prevented using vgchange or lvchange --setautoactivation n. The lvm.conf auto_activation_vol? ume_list is another way to limit autoactivation.

event autoactivation

LVM autoactivation is "event based", in which complete VGs are acti? vated in response to uevents which occur during system startup or at any time after the system has started. An old form of autoactivation was "static" in which complete VGs are activated at a fixed point dur? ing system startup by a systemd service, and not in response to events.

Event based autoactivation is driven by udev, udev rules, and systemd. When a device is attached to a machine, a uevent is generated by the kernel to notify userspace of the new device. systemd-udev runs udev rules to process the new device. Udev rules use blkid to identify the device as an LVM PV and then execute the lvm-specific udev rule for the device, which triggers autoactivation.

There are two variations of event based autoactivation that may be used on a system, depending on the LVM udev rule that is installed (found in /lib/udev/rules.d/.) The following summarizes the steps in each rule which lead to autoactivation:

69-dm-lvm-metad.rules

? device /dev/name with major:minor X:Y is attached to the machine

? systemd/udev runs blkid to identify /dev/name as an LVM PV

? udev rule 69-dm-lvm-metad.rules is run for /dev/name

? the lvm udev rule runs the systemd service lvm2-pvscan@X:Yservice

? the lvm2-pvscan service runs:

  pvscan --cache -aay --major X --minor Y

? pvscan reads the device, records that the PV is online (see pvs_on?

  line), and checks if the VG is complete.

? if the VG is complete, pvscan creates the vgs_online temp file, and

  activates the VG.

? the activation command output can be seen from systemctl status

  lvm2-pvscan*

69-dm-lvm.rules

? device /dev/name with major:minor X:Y is attached to the machine

? systemd/udev runs blkid to identify /dev/name as an LVM PV

? udev rule 69-dm-lvm.rules is run for /dev/name

? the lvm udev rule runs:

  pvscan --cache --listvg --checkcomplete --vgonline

  --autoactivation event --udevoutput --journal=output /dev/name

? pvscan reads the device, records that the PV is online (see pvs_on?

  line), and checks if the VG is complete.

? if the VG is complete, pvscan creates the vgs_online temp file, and

  prints the name of the VG for the udev rule to import:

  LVM_VG_NAME_COMPLETE='vgname'

? if the lvm udev rule sees LVM_VG_NAME_COMPLETE from pvscan, it acti?

  vates the VG using a transient systemd service named lvm-acti?

  vate-<vgname>.

? the lvm-activate-<vgname> service runs vgchange -aay --autoactivation event <vgname>

? the activation command output can be seen from journalctl -u lvm-ac?

tivate-<vgname>

pvscan options

--cache

Read the <device> arg (and only that device), and record that the PV is

online by creating the /run/lvm/pvs_online/<pvid> file containing the

name of the VG and the device for the PV.

-aay

Activate the VG from the pvscan command (includes implicit --checkcom?

plete and --vgonline.)

--checkcomplete

Check if the VG is complete, i.e. all PVs are present on the system, by

checking /run/lvm/pvs_online/<pvid> files.

--vgonline

Create /run/lvm/vgs_online/<vgname> if the VG is complete (used to en?

sure only one command performs activation.)

--autoactivation event

Inform the command it is used for event based autoactivation.

--listvg

Print the name of the VG using the device.

--udevoutput

Only print output that can be imported to the udev rule, using the udev

environment key format, i.e. NAME='value'.

--journal=output

Send standard command output to the journal (when stdout is reserved

for udev output.)

run files

Autoactivation commands use a number of temp files in /run/lvm (with

the expectation that /run is cleared between boots.)

pvs_online

pvscan --cache creates a file here for each PV that is attached. The

file is named with the PVID and contains the VG name and device infor?

mation. The existence of the file is used to determine when all PVs for a given VG are present. The device information in these files is also used to optimize locating devices for a VG when the VG is activated.

pvs_lookup

pvscan --cache creates a file here named for a VG (if one doesn't already exist.) The file contains a list of PVIDs in the VG. This is needed when a PV is processed which has no VG metadata, in which case the list of PVIDs from the lookup file is used to check if the VG is complete.

vgs_online

The first activation command (pvscan or vgchange) to create a file here, named for the VG, will activate the VG. This resolves a race when concurrent commands attempt to activate a VG at once.

static autoactivation

A static autoactivation method is no longer provided by lvm. Setting event_activation=0 still disables event based autoactivation. WARNING: disabling event activation without an alternative may prevent a system from booting. A custom systemd service could be written to run autoactivation during system startup, in which case disabling event autoactivation may be useful.

EXAMPLES

VG "vg" contains two PVs:

```
$ pvs -o name,vgname,uuid /dev/sdb /dev/sdc
  PV         VG PV UUID
  /dev/sdb   vg 1uKpaT-lFOZ-NLHX-j4jI-OBi1-QpdE-HZ5hZY
  /dev/sdc   vg 5J3tM8-aIPe-2vbd-DBe7-bvRq-TGj0-DaKV2G
```

use of --cache:

```
$ pvscan --cache /dev/sdb
  pvscan[12922] PV /dev/sdb online.
$ pvscan --cache /dev/sdc
  pvscan[12923] PV /dev/sdc online.
$ cat /run/lvm/pvs_online/1uKpaTlFOZNLHXj4jIOBi1QpdEHZ5hZY
```

```
8:16
vg:vg
dev:/dev/sdb
$ cat /run/lvm/pvs_online/5J3tM8aIPe2vbdDBe7bvRqTGj0DaKV2G
8:32
vg:vg
dev:/dev/sdc
```

use of -aay:

```
$ pvscan --cache -aay /dev/sdb
  pvscan[12935] PV /dev/sdb online, VG vg incomplete (need 1).
$ pvscan --cache -aay /dev/sdc
  pvscan[12936] PV /dev/sdc online, VG vg is complete.
  pvscan[12936] VG vg run autoactivation.
  1 logical volume(s) in volume group "vg" now active
$ cat /run/lvm/pvs_online/1uKpaTlFOZNLHXj4jIOBi1QpdEHZ5hZY
8:16
vg:vg
dev:/dev/sdb
$ cat /run/lvm/pvs_online/5J3tM8aIPe2vbdDBe7bvRqTGj0DaKV2G
8:32
vg:vg
dev:/dev/sdc
$ ls /run/lvm/vgs_online/vg
/run/lvm/vgs_online/vg
```

use of --listvg:

```
$ pvscan --cache --listvg /dev/sdb
  VG vg
$ pvscan --cache --listvg /dev/sdc
  VG vg
$ cat /run/lvm/pvs_online/1uKpaTlFOZNLHXj4jIOBi1QpdEHZ5hZY
8:16
vg:vg
dev:/dev/sdb
```

```
$ cat /run/lvm/pvs_online/5J3tM8aIPe2vbdDBe7bvRqTGj0DaKV2G
8:32
vg:vg
dev:/dev/sdc
```

use of --checkcomplete:

```
$ pvscan --cache --listvg --checkcomplete --vgonline /dev/sdb
  pvscan[12996] PV /dev/sdb online, VG vg incomplete (need 1).
  VG vg incomplete
$ pvscan --cache --listvg --checkcomplete --vgonline /dev/sdc
  pvscan[12997] PV /dev/sdc online, VG vg is complete.
  VG vg complete
```

use of --udevoutput:

```
$ pvscan --cache --listvg --checkcomplete --vgonline --udevoutput /dev/sdb
LVM_VG_NAME_INCOMPLETE='vg'
$ pvscan --cache --listvg --checkcomplete --vgonline --udevoutput /dev/sdc
LVM_VG_NAME_COMPLETE='vg'
```

use of --listlvs:

```
$ lvs -o name,devices vg
  LV    Devices
  lvol0 /dev/sdb(0)
  lvol1 /dev/sdc(0)
  lvol2 /dev/sdb(1),/dev/sdc(1)
$ pvscan --cache --listlvs --checkcomplete /dev/sdb
  pvscan[13288] PV /dev/sdb online, VG vg incomplete (need 1).
  VG vg incomplete
  LV vg/lvol0 complete
  LV vg/lvol2 incomplete
$ pvscan --cache --listlvs --checkcomplete /dev/sdc
  pvscan[13289] PV /dev/sdc online, VG vg is complete.
  VG vg complete
  LV vg/lvol1 complete
  LV vg/lvol2 complete
```