RHEL: udev rules basics

Article Number: 142 | Rating: Unrated | Last Updated: Sat, Jun 2, 2018 9:11 AM

RHEL: udev rules basics

```
# Tested on RHEL 6 & 7

# 'udev' is a mechanism for maintaining device nodes on a server. It
handles the management
# of the devices nodes through the 'udevd' daemon as well as rules
defined on the system.
# This way we can manage the creation of device nodes and their
properties and names.

# Once a device is recognized by the kernel it triggers a series of
events. First it
# populates /sys structures, then the kernel sends a uevent received
by 'udev' and,
# finally, 'udev' creates a device node for the new device or parses
the 'udev' rules files,
# under /etc/udev/rules.d/, in alphanumeric order, to decide what
action should be taken.



# To query the 'udev' database for device information or the
properties of a device from its
# sysfs representation we can use 'udevadm info':

# Export the content of the udev database:

udevadm info --export-db

# Print all sysfs properties of the specified device that can be used
in 'udev' rules to match
```

```
# the specified device. It prints all devices along the chain, up to
the root of sysfs that
# can be used in 'udev' rules.

udevadm info --attribute-walk --name=/dev/<sdc>

# For a scsi device, we may use its unique SCSI identifier



# 'udevd' watch for any changes in rules files and will automatically
apply any changes to
# those files. If that's not the case, one can ask 'udevd' to reload
the rules files
# (reloading rules does not apply any changes to already existing
devices) by:

udevadm control --reload-rules

# Request to trigger the execution of all 'udev' rules:

udevadm trigger



# Writing a simple custom rule
# ---------------------------------------------------------------------
-----------------------

# A 'udev' rule has two parts: one part that tests certain conditions
and one part that
# assigns variables (name, permissions, symbolic links, etc) when all
conditions are
# fulfilled

# 'udev' rule operators
# ----------------------
#
```

```
#   ==   Compare for equality
#   !=   Compare for inequality
#    =   Assign a value to a key
#   += Add the value to a key that holds a list of entries
#   :=   Assign a value to a key and lock the key to prevent any
further modification


# Some examples of udev rules:

vi /etc/udev/rules.d/<99-my-custom.rules>


# 1.- Create a symbolic link under /dev/mydisks/ for each matching sd
device

   SUBSYSTEM=="block", KERNEL=="sd*", SYMLINK:="mydisks/%k"

# If the new device is a block device, then we test if the internal
kernel name starts
# with 'sd'. If both match, we add 'mydisks/%k' to the list of
symbolic links to be created
# for this device



# 2.- Create a symbolic link and run a command for each sd matching
device

   KERNEL=="sd[b-d]", SUBSYSTEM=="block", SUBSYSTEMS=="scsi",
DRIVERS=="sd", SYMLINK+="shared/%k", RUN+="/usr/bin/wall
MESSAGE: shared/%k  -->  $tempnode"


ll /dev/sd*
   brw-rw---- 1 root disk 8,  0 Jul  2 17:18 /dev/sda
   brw-rw---- 1 root disk 8,  1 Jul  2 17:18 /dev/sda1
   brw-rw---- 1 root disk 8,  2 Jul  2 17:18 /dev/sda2
   brw-rw---- 1 root disk 8, 16 Jul  2 17:18 /dev/sdb
   brw-rw---- 1 root disk 8, 32 Jul  2 17:18 /dev/sdc
   brw-rw---- 1 root disk 8, 48 Jul  2 17:18 /dev/sdd
```

udevadm trigger

```
   Broadcast message from root@mynode (Wed Jul  2 17:18:44 2014):

   MESSAGE: /dev/sdd

   Broadcast message from root@mynode (Wed Jul  2 17:18:45 2014):

   MESSAGE: /dev/sdb

   Broadcast message from root@mynode (Wed Jul  2 17:18:46 2014):

   MESSAGE: /dev/sdc
```

ll /dev/shared/

```
   total 0
   lrwxrwxrwx 1 root root 6 Jul  2 17:18 sdb -> ../sdb
   lrwxrwxrwx 1 root root 6 Jul  2 17:18 sdc -> ../sdc
   lrwxrwxrwx 1 root root 6 Jul  2 17:18 sdd -> ../sdd
```

# 3.- Name a network interface bearing a given mac-address (/etc/udev/rules.d/**70-persistent-net.rules**)

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:48:a7:ef", ATTR{type}=="1", KERNEL=="eth*", NAME="my_eth0"

```
# 4.- Create a symbolic link and run a command for a device matching
a given SCSI identifier

SUBSYSTEM=="block", KERNEL=="sd?", PROGRAM=="/usr/lib/udev/scsi_id -g
-u -d /dev/%k", RESULT=="1ATA_VBOX_HARDDISK_VB0d221c89-8b845902",
SYMLINK:="shared/disk01"

ll /dev/shared
   total 0
   lrwxrwxrwx 1 root root 6 Feb  8 16:18 disk01 -> ../sdb



# 5.- Creating Oracle ASM devices under /dev/oracle/ using disks'
SCSI identifiers

ACTION=="add", BUS=="scsi",
ENV{ID_SERIAL}=="360c22lm4rr4nzf88df13aa9a0c8",
NAME="oracle/DGTEST_01", OWNER="oracle", GROUP="dba", MODE="0660"
```

Posted - Sat, Jun 2, 2018 9:11 AM. This article has been viewed 8387 times.

Online URL: http://kb.ictbanking.net/article.php?id=142