
Subject: Re: /dev/loop support inside VEs?
Posted by [phpfreak](#) on Tue, 07 Feb 2006 13:57:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Kir,

Could you please provide an example on what you mean to change? I changed vzfs to simfs and mkvzfs to mkfs on mine, but did not have much luck.

```
/etc/sysconfig/vz-scripts/vps.mount
#!/bin/bash
#
# This script is global and executed for every VPS at startup time.
# We are going to create and mount a temp area with nosuid, nodev and noexec,
# which will have vzquota configured and running.

# Current issues:
# 1) vzquota accepts only numeric and does it in a very weird way. Details below.
# 2) not clear how to handle on->off and off->on changes for tmp area--i.e. what to do with files
# under /tmp and /var/tmp.
# it's possible to move files back and forth on mount/umount stage--i.e.
#
# mv tmp temptmp
# mount tmparea
# tar -cf - -C temptmp . | tar xpf - -C tmp
#
# on mount and opposite action on umount but it may take considerable time - we have quotas
already
# running, it's copying across mountpoints etc.
# 3) perhaps tmp should be added to /etc/fstab
# 4) completely unclear what to do with second-level quotas.

# script works with $VEID and $VE_CONFIGFILE vars which are passed as environment
# variables. All the rest can be defined
# a) in /etc/sysconfig/vz as a system-wide
# and b) in VE config file.

# tmp sizes/limits
VPSTMP_BLOCKS=$((150*1024))
VPSTMP_INODES=2000

# tmp 'path' - we might want have it outside
# of /vz
TMPPATH="/vz/private"
VPSTMP="$VEID-temparea"

# currently service VPS just doesn't work right
# with a dedicated nosuid / noexec TMP.
```

```

if [ $VEID -eq 1 ]; then
    exit 0
fi

# source configs.
if [ -f /etc/sysconfig/vz ]; then
    . /etc/sysconfig/vz
else
    exit 1
fi

if [ -f $VE_CONFFILE ]; then
    . $VE_CONFFILE
else
    exit 1
fi

# a special var from either global file or VPS config.
if [ -z "$VPS_TMP_AREA" ]; then
    # TMP area not configured in neither config.
    exit 0
fi

if [ "$VPS_TMP_AREA" != "yes" -a "$VPS_TMP_AREA" != "YES" ]; then
    # TMP area is disabled in either config
    exit 0
fi

# after sourcing configs we might have blocks/inodes in limit:barrier form
# and have to handle it. Perhaps we need to check that soft < hard here.

if [ "$VPSTMP_BLOCKS" = "${VPSTMP_BLOCKS/./}" ]; then
    VPSTMP_BLOCKS_SOFT=$VPSTMP_BLOCKS
    VPSTMP_BLOCKS_HARD=$VPSTMP_BLOCKS
else
    VPSTMP_BLOCKS_SOFT=${VPSTMP_BLOCKS%%.*}
    VPSTMP_BLOCKS_HARD=${VPSTMP_BLOCKS##*.}
fi

if [ "${VPSTMP_INODES}" = "${VPSTMP_INODES/./}" ]; then
    VPSTMP_INODES_SOFT=$VPSTMP_INODES
    VPSTMP_INODES_HARD=$VPSTMP_INODES
else
    VPSTMP_INODES_SOFT=${VPSTMP_INODES%%.*}
    VPSTMP_INODES_HARD=${VPSTMP_INODES##*.}
fi

```

```
# it seems that vzquota not only doesn't work with non-numeric but also silently
# removes non-numeric chars from supplied , without reporting errors.
# this indeed is very unfortunate since we have to use something like $00001
# instead of $VEID-tmparea for --otherwise there're some weird interaction
# between VPS and temparea quotas.
```

```
### WARNING!!!!!!#####
```

```
# VPS ID can not be more than 2^32-1, if you use "big" IDs for VPSs, you have to
# modify a var below to have VPSTMP_QUOTAID below the VPS ID "limit"
# (this limit also applies to quota IDs)
```

```
VPSTMP_QUOTAID=${VEID}1111
```

```
# other constants
```

```
# VZ_PRIVATE=/vz/private
```

```
# strip trailing slashes from TMPPATH
```

```
TMPPATH=${TMPPATH%%/?}
```

```
# extra sanity check
```

```
if [ "$TMPPATH/$VPSTMP" = "/" ]; then
    exit 1
fi
```

```
# if we don't have "vzfs filesystem" for the temp
# area, we have to create it, and init quota on it.
```

```
if [ ! -d "$TMPPATH/$VPSTMP" ]; then
```

```
    mkvzfs $TMPPATH/$VPSTMP
```

```
    RETVAL=$?
```

```
    if [ $RETVAL -ne 0 ]; then
```

```
        # some logging?
```

```
        exit $RETVAL
```

```
    fi
```

```
    vzquota init $VPSTMP_QUOTAID -p $TMPPATH/$VPSTMP \
```

```
        -c /var/vzquota/quota.$VPSTMP_QUOTAID \
```

```
        --block-softlimit $VPSTMP_BLOCKS_SOFT \
```

```
        --block-hardlimit $VPSTMP_BLOCKS_HARD \
```

```
        --block-exptime 0 \
```

```
        --inode-softlimit $VPSTMP_INODES_SOFT \
```

```
        --inode-hardlimit $VPSTMP_INODES_HARD \
```

```
        --inode-exptime 0
```

```
    RETVAL=$?
```

```
    if [ $RETVAL -ne 0 ]; then
```

```
        # some logging?
```

```
        exit $RETVAL
```

```
    fi
```

```
fi
```

```

# turning quota on.
vzquota on $VPSTMP_QUOTAID
RETVAL=$?
if [ $RETVAL -ne 0 ]; then
    # some logging
    exit $RETVAL
fi

# OK, assuming that everything is done. Now we need to mount tmp.
if [ ! -d "$TMPPATH/$VPSTMP" ]; then
    # something really is broken.
    exit 1
else
    mount -t vzfs \
        -o noatime,nosuid,noexec,nodev,rw,/vz/template:$TMPPATH/$VPSTMP \
        /vz/template:$TMPPATH/$VPSTMP $VE_ROOT/tmp
    RETVAL=$?
    if [ $RETVAL != 0 ]; then
        # some logging
        exit $RETVAL
    fi
    # we want tmp to have 1777 mode
    chmod 1777 $VE_ROOT/tmp
fi

# if we are here, everything is good so far
# we want to make /var/tmp to be symlink to /tmp.

if [ ! -L $VE_ROOT/var/tmp ]; then
    rm -rf $VE_ROOT/var/tmp
    ln -s /tmp $VE_ROOT/var/tmp
fi

exit 0

/etc/sysconfig/vz-scripts/vps.umount

#!/bin/bash
#
# this script is global and executed for every VPS at stop time
# we're going to umount a temp area and stop vzquota for it.

# script works with $VEID and $VE_CONFFILE vars which are passed as environment
# variables. All the rest can be defined
# a) in /etc/sysconfig/vz as a system-wide

```

and b) in VE config file.

```
TMPPATH="/vz/private"  
VPSTMP="$VEID-temparea"
```

```
# currently service VPS just doesn't work right  
# with a dedicated nosuid / noexec TMP.
```

```
if [ $VEID -eq 1 ]; then  
    exit 0  
fi
```

```
# source configs.  
if [ -f /etc/sysconfig/vz ]; then  
    . /etc/sysconfig/vz  
else  
    exit 1  
fi
```

```
if [ -f $VE_CONFFILE ]; then  
    . $VE_CONFFILE  
else  
    exit 1  
fi
```

script is really simple and most likely should be changed completely

```
VPSTMP_QUOTAID=${VEID}1111
```

```
if grep -q $VPSTMP /proc/mounts; then  
    umount $VE_ROOT/tmp  
    RETVAL=$?  
    if [ $RETVAL -ne 0 ]; then  
        # some logging?  
        # do we need 'umount -f' here?  
        exit $RETVAL  
    fi  
    vzquota off $VPSTMP_QUOTAID  
    RETVAL=$?  
    if [ $RETVAL -ne 0 ]; then  
        # some logging?  
        exit $RETVAL  
    fi  
fi  
  
exit 0
```

Also, I am assuming we can put this inside the /etc/sysconfig/vz-scripts/VEID.conf:

```
VPS_TMP_AREA="yes"
```
