OpenNebula - Bug #5116
Ceph mkimage fails with incompatible rbd features

04/19/2017 01:54 PM - Gerben Meijer

Status:	Pending	**Start date:**	04/19/2017
Priority:	Normal	**Due date:**	
Assignee:	Vlastimil Holer	**% Done:**	0%
Category:	Drivers - Storage	**Estimated time:**	0.00 hour
Target version:	Release 5.6		
Resolution:		**Pull request:**	
Affected Versions:	OpenNebula 5.2, OpenNebula 5.4		

Description

mkimage in the Ceph TM driver does an 'rbd map' to create a swap filesystem.

This does not work with all kernels; for example:

```
[2451939.802702] rbd: image one-10-3-0: image uses unsupported features: 0x38
```

A possible workaround is to depend on libguestfs and use guestfish to create it. Basic example:

```
if [ "$FSTYPE" = "swap" ]; then
    export LIBGUESTFS_BACKEND=direct
    ( echo set-attach-method appliance
      echo add-drive /dev/null
      echo config -set drive.hd0.file=rbd:$RBD_SOURCE
      echo run
      echo mkswap /dev/sda label:swap ) | $SUDO /usr/bin/guestfish
fi
```

Note that this will also require the addition of /usr/bin/guestfish to sudoers.

I think it's the most elegant solution since it becomes kernel and distro independent.

History

#1 - 04/19/2017 04:43 PM - Roy Keene

A simpler solution is to use "qemu-nbd" to create an NBD that corresponds to the RBD. If QEMU is compiled with RBD support then the QEMU tools have the ability to map RBD devices.

Failing that, it can be done by creating a sparse file and writing the header to it, then reading that header into a new RBD image:

```
(
    tmpfile="$(mktemp)" && \
        dd if=/dev/zero of="${tmpfile}" bs=$[1024*1024] seek=$[${SIZE} - 1] count=1 2>/dev/null && \
        mkswap "${tmpfile}" && \
        dd if="${tmpfile}" bs=$(getconf PAGESIZE) count=1 2>/dev/null | rbd import - $RBD && \
        rbd resize --size ${SIZE}M $RBD
    rm -f "${tmpfile}"
)
```

#2 - 04/19/2017 04:54 PM - Roy Keene

- File opennebula-5.2.1-mkswapsparse.diff added

#3 - 04/19/2017 06:46 PM - Roy Keene

- File mkswap-qemu.sh added

Also attached is a wrapper script called "mkswap-qemu" which uses "qemu-img" to handle both RBD and QCOW2 (and any other format QEMU supports, if properly extended) and run "mkswap"

#4 - 08/28/2017 03:55 PM - Javi Fontan

- Target version set to Release 5.6

- Affected Versions OpenNebula 5.4 added

I'm marking this to be done in the next major release.

#5 - 09/04/2017 03:39 PM - Javi Fontan

- Assignee set to Vlastimil Holer

#6 - 09/21/2017 07:08 AM - Tobias Rehn

We are also experiencing this issue also. I don't think that it is a good idea to create the swap file locally. For virtualization host with small HDDs such a way can fill up the HDD and lead to further issues.

We are using the following workaround:

```
if [ "$FSTYPE" = "swap" ]; then
    $SUDO $RBD map $RBD_SOURCE --image-feature layering || exit \$?
    $SUDO $MKSWAP -L swap /dev/rbd/$RBD_SOURCE
    $SUDO $RBD unmap /dev/rbd/$RBD_SOURCE
fi
```

By only using the rbd feature "layering" it works without any issues on older kernels like 3.10 (rhel7 / centos7). Most of the newer features like "deep-flatten", "fast-diff" or "object-map" are only available in newer kernels. Even the latest mainline kernel 4.13 does not support all features. This is because the krbd features are lacking behind the features of rbd.

When using latest ceph luminious you want to use all new features for your images as these have enormous speed enhancements for clones, etc. But you will never ever clone a swap disk - so layering is enough for this type of storage disk.

#7 - 09/21/2017 07:10 AM - Tobias Rehn

I had a failure in my workaround:

```
MKIMAGE_CMD=$(cat <<EOF
    export PATH=/usr/sbin:/sbin:\$PATH
```

```
    if [ "$FSTYPE" = "swap" ]; then
        $RBD create $FORMAT_OPT $RBD_SOURCE --size ${SIZE} --image-feature layering || exit \$?
        $SUDO $RBD map $RBD_SOURCE || exit \$?
        $SUDO $MKSWAP -L swap /dev/rbd/$RBD_SOURCE
        $SUDO $RBD unmap /dev/rbd/$RBD_SOURCE
    fi

    $RBD create $FORMAT_OPT $RBD_SOURCE --size ${SIZE} || exit \$?

EOF
)
```

#8 - 09/21/2017 08:26 AM - Tobias Rehn

After some further testing I have come the result that I see no advantage from distinguishing between swap or normale volatile disk. I thought the swap disk were automatically mounted but that is not the case and I see no advantage. So we working with the following:

```
MKIMAGE_CMD=$(cat <<EOF
    export PATH=/usr/sbin:/sbin:\$PATH
    $RBD create $FORMAT_OPT $RBD_SOURCE --size ${SIZE} || exit \$?
EOF
)
```

Files

opennebula-5.2.1-mkswapsparse.diff	1.11 KB	04/19/2017	Roy Keene
mkswap-qemu.sh	993 Bytes	04/19/2017	Roy Keene