Windows Virtuals

Windows 11

Windows 11 requires TPM and Secure Boot. See Install Windows 11 Virtual for detailed instructions

Shutdown and Reboot don't work

If you can not issue the command

```
virsh shutdown domname
```

where *domname* is a running domain, the problem may be that the QEMU guest agent is either not installed on your Windows virtual, or the configuration for the domain on your hypervisor is not set up.

This assumes you are using the QEMU guest agent which is installed by default when you use the virtio-win drivers.

First, verify that is the problem by issuing

```
virsh shutdown domname --mode agent
```

where domname is, again, a running virtual. This forces virsh to try the shutdown only using the QEMU agent, and it will give you an error if it can not do so. On my tests, the error was:

```
error: argument unsupported: QEMU guest agent is not configured
```

which translates to *Your configuration file does not have the necessary information to do this.*

In this case, edit the configuration manually with the command

```
virsh edit domname
```

then find the line which reads </device> (the final line of the device section). Insert the following block **above** that line (keeping it in the <device> </device> section.

```
    <channel type='unix'>
      <source mode='bind' path='/var/lib/libvirt/qemu/<guest-name>.agent'/>
      <target type='virtio' name='org.qemu.guest_agent.0'/>
    </channel>
```

NOTE change <*guest-name*> to the name of the virtual. Do not leave it as <*guest-name*>.

Save your changes.

Now, log into the virtual and manually shut it down (do not restart, shut down). Back on the

hypervisor, start the virtual back up.

Now, running

```
virsh shutdown domname
```

should properly shut down the domain.

virtual will not reboot

I have found that some (not all) Windows virtuals will not restart/reboot, either from within the running virtual, or from the virsh command line. Instead, they shut down.

Since this causes many issues, we wrote a Perl script as a work around. The script simply checks to see if virtuals are up and, if not, starts them (with the virsh start command). In our case, we decided that this would occur either when doing maintenance on the machine, or overnight when the virtual decides to reboot itself, so a 10 minute, maximum, downtime was acceptable.

checkVirtuals

```
#! /usr/bin/env perl

use strict;
use warnings;

# list of virtuals which should be turned on if they
# are found to be down. This is the same as that returned
# by virsh list
my @servers = (
   'win1',
   'win2',
   'win3'
   );

# command to start a virtual
my $virsh = '/usr/bin/virsh start ';

# get a list of running virtuals and put them into
# $output
my $output = `virsh list`;

# check each virtual we are monitoring
foreach my $server ( @servers ) {
   if ( $output =~ m/$server/ ) { # the virtual is running
      # remove the flag file
      unlink "/tmp/$server.down" if  -e "/tmp/$server.down";
   } else { # virtual is not running
      if ( -e "/tmp/$server.down" ) { # it was down on the previous
pass
```

```
            print "$server has been down for a while, starting back up\n";
            # so bring it back up
            `$virsh $server`;
            # and remove the flag file
            unlink "/tmp/$server.down";
         } else { # just went down
            # create a flag file so the next pass will start it back up
            `touch /tmp/$server.down`;
         }
      }
}

1;
```

Every time this script is run, it will check each of the virtuals it is monitoring to see if it is up. The first time it finds one that is not up, it will create a flag file in /tmp, named ***virtualname***.*down*.

When it finds it is down the second time, it will start it back up.

The following cron file, stored in /etc/cron.d (Devuan/Debian) will run the script every 5 minutes. The minimum down time will be 5 minutes, 1 second, and the maximum downtime will be 10 minutes.

checkVirtuals.cron

```
# Check status of virtuals on this machine and, restart any if down.
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
# following may be changed to send failure reports to a different e-
mail
# account
MAILTO=root

# check running virtuals every 5 minutes

# m  h dom mon dow   user    command
*/5 *  *   *   *    root    perl /opt/scripts/virtuals/checkVirtuals

# EOF
```

Save this file to /etc/cron.d/checkVirtuals (no .cron after it). Edit the */5 to determine how many minutes to do the check, and edit the command to point to wherever you put the checkVirtuals Perl script.

Links

- https://access.redhat.com/solutions/732773
- https://github.com/virt-manager/virt-manager/issues/271

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/unix/virtualization/virtlib/winvirtuals

Last update: **2025/01/18 21:32**

