

openQA Project - action #28441

Nested virtualization for openqa instance fails

2017-11-27 10:12 - ktsamis

Status:	Resolved	Start date:	2017-11-27
Priority:	Normal	Due date:	
Assignee:	okurz	% Done:	0%
Category:	Support	Estimated time:	0.00 hour
Target version:	future		
Difficulty:			

Description

I created a VM with SLES12SP2 to have as an openqa instance, I followed internal instructions for installation and configuration but they are identical to the public [instructions](#).

Because I was using a VM as the instance, I needed to enable nested virtualization in the host system for the workers to exist. The hypervisor is the default workstation we have in SUSE, a Dell Precision Tower 5810 with this Processor: Intel® Xeon® Processor E5-1620 v4 (4C, 3.5GHz, 3.8GHz Turbo, 2400MHz, 10MB, 140W).

This was set up correctly:

```
cat /sys/module/kvm_intel/parameters/nested
Y
```

After cloning a passing job from openqa.suse.de I tried to run it on my instance and I got these errors:

```
[Mon Nov 27 10:03:57 2017] [9197:debug] Error connecting to host <localhost>: IO::Socket::INET: connect: Connection refused
DIE can't open hmp at /usr/lib/os-autoinst/backend/qemu.pm line 792.
```

```
at /usr/lib/os-autoinst/backend/baseclass.pm line 80.
  backend::baseclass::die_handler('can't open hmp at /usr/lib/os-autoinst/backend/qemu.pm line ...') called at /usr/lib/os-autoinst/backend/qemu.pm line 792
  backend::qemu::start_qemu('backend::qemu=HASH(0x69b5728)') called at /usr/lib/os-autoinst/backend/qemu.pm line 103
  backend::qemu::do_start_vm('backend::qemu=HASH(0x69b5728)') called at /usr/lib/os-autoinst/backend/baseclass.pm line 257
  backend::baseclass::start_vm('backend::qemu=HASH(0x69b5728)', undef) called at /usr/lib/os-autoinst/backend/baseclass.pm line 75
  backend::baseclass::handle_command('backend::qemu=HASH(0x69b5728)', 'HASH(0x6c042e8)') called at /usr/lib/os-autoinst/backend/baseclass.pm line 436
  backend::baseclass::check_socket('backend::qemu=HASH(0x69b5728)', 'IO::Handle=GLOB(0x6851440)') called at /usr/lib/os-autoinst/backend/qemu.pm line 1000
  backend::qemu::check_socket('backend::qemu=HASH(0x69b5728)', 'IO::Handle=GLOB(0x6851440)', 0) called at /usr/lib/os-autoinst/backend/baseclass.pm line 208
  eval {...} called at /usr/lib/os-autoinst/backend/baseclass.pm line 156
  backend::baseclass::run_capture_loop('backend::qemu=HASH(0x69b5728)') called at /usr/lib/os-autoinst/backend/baseclass.pm line 129
  backend::baseclass::run('backend::qemu=HASH(0x69b5728)', 5, 8) called at /usr/lib/os-autoinst/backend/driver.pm line 85
  backend::driver::start('backend::driver=HASH(0x4d37838)') called at /usr/lib/os-autoinst/backend/driver.pm line 48
  backend::driver::new('backend::driver', 'qemu') called at /usr/bin/isotovideo line 211
  main::init_backend() called at /usr/bin/isotovideo line 280
[Mon Nov 27 10:03:58 2017] [9197:debug] waitpid for 9203 returned 9203
last frame
[Mon Nov 27 10:03:58 2017] [9197:debug] QEMU: Could not access KVM kernel module: No such file or directory
[Mon Nov 27 10:03:58 2017] [9197:debug] QEMU: failed to initialize KVM: No such file or directory
[Mon Nov 27 10:03:58 2017] [9197:debug] sending magic and exit
[Mon Nov 27 10:03:58 2017] [9190:debug] received magic close
failed to start VM at /usr/lib/os-autoinst/backend/driver.pm line 128.
[Mon Nov 27 10:03:58 2017] [9190:debug] awaiting death of commands process
```

```
[Mon Nov 27 10:03:58 2017] [error] can_read received kill signal at /usr/lib/os-autoinst/myjsonrpc.pm line 89.
```

I tried to load the kernel module manually in the instance:

```
# modprobe kvm_intel
modprobe: ERROR: could not insert 'kvm_intel': Operation not supported
```

My physical CPU on the host system is (1/8 cores):

```
# cat /proc/cpuinfo
processor       : 0
vendor_id     : GenuineIntel
cpu family    : 6
model        : 63
model name    : Intel(R) Xeon(R) CPU E5-1620 v3 @ 3.50GHz
stepping     : 2
microcode    : 0x3a
cpu MHz      : 3600.008
cache size   : 10240 KB
physical id  : 0
siblings    : 8
core id     : 0
cpu cores   : 4
apicid     : 0
initial apicid : 0
fpu        : yes
fpu_exception : yes
cpuid level : 15
wp         : yes
flags      : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
            dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon peb
            s bts rep_good
            nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2
            ssse3 sdbg fma cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes x
            save avx f16c r
            drand lahf_lm abm ida arat epb pln pts dtherm tpr_shadow vnmi flexpriority ept vpid fsgsbase tsc_a
            djust bmi1 avx2 smep bmi2 erms invpcid cqm xsaveopt cqm_llc cqm_occup_llc
            bugs
            bogomips      : 6984.40
            clflush size  : 64
            cache_alignment : 64
            address sizes  : 46 bits physical, 48 bits virtual
            power management:
```

I checked "Copy host CPU Configuration" in the virt-manager and the Model was set as "Haswell-noTSX".

I know this is not an openqa specific issue, more like a kvm nested virtualization issue but I need help to fix this for openqa and if someone has already installed their instance in a VM maybe they can help me fix it. If this is not possible (VM as openqa instance) or you know of a workaround please let me know. Otherwise we need to document that the openQA instance needs to be a physical machine.

History

#1 - 2017-11-27 10:31 - coolo

- Category set to Support

- Target version set to future

perhaps you better post it to a mailing list with virtualization experts instead of posting it in the issue backlog?

#2 - 2017-11-28 06:38 - okurz

- Status changed from New to Feedback

- Assignee set to okurz

nested virt is possible in general. Can you run `sudo virt-host-validate` on a virtual machine that you start manually on that host to check?

#3 - 2017-12-13 01:00 - AdamWill

I have used a VM (to be specific, an Openstack instance) as a worker host twice in the past. It does work. The capabilities of the CPU in the 'host' VM are very important, though. Specifically, IIRC, you need to make sure that 'vmx' is in the CPU flags for the host VM (not just the physical box itself). If it isn't, that's your problem; you need to fiddle with the host VM configuration until it is.

#4 - 2017-12-13 09:15 - okurz

- *Status changed from Feedback to Resolved*

no response on support task. That could mean that the problem does not appear anymore :)

#5 - 2017-12-13 10:00 - ktsamis

Hi, sorry, I was busy with other things, so I didn't check this. Leave it to resolved and I will reopen when I check and if it doesn't work for me. Sorry for the delay

#6 - 2018-06-15 19:08 - okurz

- *Target version changed from future to future*