Windows VM in KVM/QEMU on headless Ubuntu server

without using libvirt

install kvm

```
apt-get install kvm qemu
```

now create a disk image

```
qemu-img create -f raw /vm/windows.img 100G
```

(note: creating a raw image has severale advantages: if your filesystem supports sparse files, it only uses the actually used space of your virtual disk on your physical disk.. all journalling filesystems that use inodes support that, so ext4 for example works fine. second you can easily mount it using mount -o loop <imgfile> <mountpoint> at any time. however, it does not support snapshots, use qcow2 if you need snapsots or if your filesystem does not support sparse files)

in my case i had a windows 7 cd that i installed from.. if your server dos not have a cd rom you can also us an iso image instead.

since I want to use virtio for fast hdd emulation, we also need the virtio drivers

```
cd /vm
wget
https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/stable-virtio/virtio-win.iso
```

run qemu and install windows. access the display via VNC

```
qemu-system-x86_64 -enable-kvm -k de-ch -name windows -vnc :1 -cdrom
/dev/cdrom -drive file=/vm/virtio-win.iso,media=cdrom -boot d -drive
file=/vm/windows.img,if=virtio,format=raw,index=0 -m 4096
```

now connect with your favourite vnc client to the ip of your server and install windows. you need to load the derivers from the second cdrom that is emulated. make sure you choose the correct virtio storage driver for your windows version. it may load the wrong one if you just select the root directory and let the installer search for it, resulting in an unstable windows vm

autostart on boot / init script

once the installation is done, shut down your windows vm. it is now time to make it autostart upon boot of your linux machine.. in order to do that i used Thomas Martin's kvm-simple-init of which i keep a local copy (dated 2016-01-23)

this script will also send a shutdown command to all vms once the server shuts down, so no need to manually do that in order to gracefully shutdown your vm's

bridged network

by default qemu uses nat-ed network, so that the vm is in its own network and the vm host acts as a nat router. if you want your vm to be part of your network like any physical machine you have connected to your swich you can use a bridged connection..

you need to install bridge-utils to be able to create a bridge network device

```
 apt-get install bridge-utils
```

now change your `/etc/network/scripts` file and basically replace eth0 with br0 and then add three lines for the bridge config. here the end result on mine:

```
auto br0
iface br0 inet static
    address 192.168.168.1
    netmask 255.255.255.0
    network 192.168.168.0
    broadcast 192.168.168.255
    gateway 192.168.168.254
    bridge_ports eth0
    bridge_stp off
    bridge_maxwait 5
```

Notice: you won't need any block for eth0 after you defined the bridge interface.

now either reboot your machine or try to get your eth0 down using `ifconfig` and then get br0 up using `ifup` however be careful when you are doing this all remoetely!! in that case make really really sure you got your config right or even better, setup a backdoor, make sure your remote management module is accessible, whatever.. and then reboot your server.. and hope :)

now in your qemu command line you ened to add the parameters

```
-net nic,model=virtio-net-pci -net tap
```

the default qemu-ifup script in /etc/ will do the rest for you.

Notice you will need virtio drivers for the nic to work. you can leave the `model=` option away on the first boot in order to download the virtio iso and install the drivers before switching over to virtio. however, using the e1000 (default if no model is given) is not recommended for windows VM's as this model has a tendency to crash and lose connection under heavy use. I've experienced this first hand on a CCTV server that had to monitor a couple of cameras. once the nic crashes, one has to go to network settings in windows and disable, then enable the card and it will be back online. I followed the suggestions in a Proxmox forum thread and switched to virtio.

example config

here is an example configuration for a windows 7 machine using the above mentioned init script.. if you use another init script or run kvm manually you can simply use all the cli options listed below as an example of what you might want to pass on to qemu.. also take a look at the man page it is very informative and helpful!

```
# Should be unique among all VMs
MONITOR_PORT=5801

# Not mandatory, but useful to keep it in a distinct variable
VNC_DISPLAY=1

# KVM parameters
KVM_OPTS="\
-enable-kvm \
-k de-ch \
-name windows \
-drive file=/vm/windows.img,if=virtio,format=raw,index=0 \
-m 4096 \
-net nic,model=virtio-net-pci \
-net tap \
-vnc :$VNC_DISPLAY \
-monitor tcp:127.0.0.1:$MONITOR_PORT,server,nowait \
-monitor vc \
-daemonize \
```

Windows 10 upgrade

I had a windows 7 license laying around so i used that but wanted to upgrade to win 10. unfortunately the update crashed with error

```
0xC1900101 - 0x20017 The installation failed in the SAFE_OS phase with an
error during BOOT operation
```

this was easily fixed by adding a

```
-cpu core2duo
```

to my options in the above config file. keep in mind you need to stop your VM and start it again for this change to be activated.. I first removed this setting again after the update and windows 10 ran fine, but when it installed an automatic update and i accidently killed the vm during the installation it wouldn't boot anymore and it repeatedly showed a BSOD saying there was a SYSTEM_THREAD_EXCEPTION_NOT_HANDLED. at first i tried fixing my windows installation using a wind10 dvd but that was not successfull as i couldn't even boot from the dvd.. when i ran out of options i re-enabled the core2duo cpu and voilà, it bootet straight away, finished its updates and we're all good :)

that brings me to another note regarding windows vm's:

Disable Windows Updates on VM

when the host system is shut down, akk kvm vm's will be sent a acpi power down message to gracefully shut down. the problem is, if the vm does not support this feature or crashes during shutdown, the host system will in turn hang on its shutdown and you might loose control over it when you connect from remotely.. so Thomas set a timout for the shutdown command to complete and after this timeout, his init script will simply switch the vm off. this is very good in order to keep the host system under control but it can badly harm especially Windows VM's as windows likes to install updates during the shutdown of a machine and therefore can take literally hours to shut down.

there are two solutions to this problem.. an easy one and a more complex one:

1. Easy: disable automatic updates and do your updates manually so you can reboot your VM and give it all the time it wants to complete the updates
2. Complex: use the qemu monitor command `screendump <filename>` then run some command line ocr across the screen dump and see what it is actually doing.. using this output, prolong the timeout in the init script..

for once i decided to go the easy way.. the problem about the complex solution is, that this only works once you know what to look for in the screenshot.. and who knows what microsofts “i am updating, don't you turn me off” message might look like in the future.. in times of error messages like “something went wrong” my expectations aren't too high regarding a consisten message in that stage.. further more, my host system is improtant to me, it does lots of thing sand my windows vm is only used to do one single nice-to-have job (OCR on PDF Files) which i can easily live without for a moment.. so the host system has absolute priority over the windows VM and therefore i don't want to wait hours for the windows machine to complete its updates and not have any of my other other services (files, mails etc.) available at that time..

windows 10 - disable updates

windows 10 gives you a hard time when it comes to disabling updates.. it really insists on doing some of its updates automatically.. in windows 10 pro you can set some goup policies and disable automatic updates but in windows 10 home you need to work around this issue by setting your network connection to be a “metered connection” so windows won't download updates over it.

Pro - set group policies

since i am using a windows 7 pro on my vm (i upgraded from a 7 ultimate i had laying around which turned into 10 pro upon the upgrade) i will disable auto updates using the official way:

- start → search for “edit group policy”
- navigate to `Computer Configuration\Administrative Templates\Windows Components\Windows Update`

- doubleclick `configure automatic updates` and switch it to “disabled”

From:
http://wiki.psuter.ch/ - **pswiki**

Permanent link:
http://wiki.psuter.ch/doku.php?id=windows_vm_in_kvm_on_headless_ubuntu_server

Last update: **09.04.2022 01:22**