

Virtualize Windows 10, convert to UEFI, and install virtio drivers on KVM/qemu

1. Let's assume `/dev/sdb` houses your bare metal installation:

```
# qemu-img convert -O qcow2 /dev/sdb ~dude/kvm/disks/win10.qcow2
# chown dude ~dude/kvm/disks/win10.qcow2
```

2. Just in case, create a snapshot:

```
$ qemu-img snapshot -c b4_firstboot ~/kvm/disks/win10.qcow2
```

3. Create `~/kvm/bin/win10.sh` and reassemble the actual hardware as close as possible, e.g.:

```
#!/bin/sh

VM_CPUS=2
VM_MEM=8192
VM_FILE_CACHE0="none"
VM_FILE_FORMAT0="qcow2"
VM_FILE_TYPE0="ide"
VM_NET_TYPE0="e1000"
VM_CPU="host,kvm=off"
VM_SOUNDHW="hda"

. "`dirname "$0"`/../lib/kvmlib.sh"
```

4. Start the VM:

```
$ sh ~/kvm/bin/win10.sh start
```

5. If there is no [BSOD](#), log into the console, and start `cmd.exe` as Administrator

6. Convert the disk to GPT and UEFI:

```
C:\Windows\System32>mbr2gpt /allowfullOS /disk:0 /validate
C:\Windows\System32>mbr2gpt /allowfullOS /disk:0 /convert
```

7. Shutdown the guest operating system

8. Add UEFI to `~/kvm/bin/win10.sh`:

```
#!/bin/sh
```

```

VM_CPUS=2
VM_MEM=8192
VM_FILE_CACHE0="none"
VM_FILE_FORMAT0="qcow2"
VM_FILE_TYPE0="ide"
VM_NET_TYPE0="e1000"
VM_CPU="host,kvm=off"
VM_SOUNDHW="hda"
VM_EXTRA="-drive if=pflash,format=raw,readonly,file=/usr/share/edk2-ovmf/OVMF_CODE.fd"

. "`dirname "$0"`/../../lib/kvmlib.sh"

```

9. Grab the latest virtio drivers from <https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/latest-virtio/virtio-win.iso>. See also <https://docs.fedoraproject.org/en-US/quick-docs/creating-windows-virtual-machines-using-virtio-drivers/>
10. Start the VM with that ISO inserted:

```
$ sh ~/kvm/bin/win10.sh start -cdrom ~/Downloads/virtio-win.iso
```

11. Install these drivers by right-clicking and selecting *Install* for each *.inf file in the respective *w10|amd64* subdirectory:
 - o vioscsi
 - o viostor
 - o NetKVM
 - o vioserial
 - o Balloon
12. Once again, start *cmd.exe* as Administrator, and instruct Windows to boot to safe mode:

```

C:\Windows\System32>bcdedit /set {default} safeboot minimal
C:\Windows\System32>shutdown /p

```

13. Create a small disk image with virtio:

```
$ qemu-img create -f qcow2 ~/kvm/disks/tmp.qcow2 1G
```

14. Attach that disk in *~/kvm/bin/win10.sh*:

```

VM_FILE_CACHE1="none"
VM_FILE_FORMAT1="qcow2"
VM_FILE_NAME1="/home/dude/kvm/disks/tmp.qcow2"
VM_FILE_TYPE1="virtio"

```

15. Start the VM again, having the virtio drivers being picked up
16. Launch *cmd.exe* as Administrator, and disable safe mode again:

```

C:\Windows\System32>bcdedit /deletevalue {default} safeboot
C:\Windows\System32>shutdown /p

```

17. Remove tmp.qcow2 from `~/kvm/bin/win10.sh`, and change both the real harddisk and the network card to virtio, which is the default of [kvm.lib.sh](#):

```
#!/bin/sh

VM_CPUS=2
VM_MEM=8192
VM_FILE_CACHE0="none"
VM_FILE_FORMAT0="qcow2"
VM_CPU="host,kvm=off"
VM_SOUNDHW="hda"
VM_EXTRA="-device virtio-scsi -drive if=pflash,format=raw,readonly,file=/usr/share/edk2-ovmf/OVMF_CODE.fd"

. "`dirname "$0"`/../lib/kvmlib.sh"
```

18. Finally, boot that VM again

19. If everything is ok, shut down the VM and remove the snapshot:

```
$ qemu-img snapshot -d b4_firstboot ~/kvm/disks/win10.qcow2
```