Migrate WINDOWS 10 Guest to WINDOWS 11 in KVM (02-2022)

Warning: You loose your Activation Key by this procedure!

- My Prerequisite:
  Host: Opensuse Tumbleweed Host with KVM/QEMU installed with YAST
  "Install Hypervisor and Tools"

  ```
  pinot:/home/ # inxi -Fx
          Host: pinot Kernel: 5.16.4-1-vanilla x86_64 Desktop: KDE Plasma 5.23.5
          Distro: openSUSE Tumbleweed 20220205
          Mobo: ASUSTeK model: TUF GAMING B550-PLUS v: Rev X.0x
          CPU: Info: 6-Core model: AMD Ryzen 5 5600G with Radeon Graphics
          Memory: RAM: total: 14.91 GiB used: 2.04 GiB (13.7%)
          Drives: Local Storage: total: 1.02 TiB used: 227.75 GiB (21.8%)
  ```

  Guest: Windows 10, Disksize:50 GB *qcow2 , MBR-Boot(WIN+R: msinfo32.exe,
  BIOS-Mode: Legacy), 1 CPU Core, No crypto chip TPM is used
  4 obstacles for updating to Windows 11

- We tackle MBR-Boot , only 1 CPU-core obstacle and TPM-Chip:
  - Do as root (we clone our win10):
    ```
    cd /var/lib/libvirt/images/
    cp win10.qcow2 win11.qcow2
    chown qemu:qemu win11.qcow2 (focus on file permissions)
    ```
  - We create a new VM with *virt-manager* with disk `win11.qcow2`
    (use parameters of the original VM)
  - Boot your new VM, get a Console as administrator and execute
    ```
    mbr2gpt.exe /allowFullOS /disk:0 /convert
    shutdown
    ```
    Now this VM is unusable!
  - Take *virt-manager* to delete the VM but NOT the disk-file (important!)
    Take *virt-manager* to create a new VM with `win11.qcow2` config with more than 1 CPU, 8GB RAM and choose: edit configuration before installation
    (because you changed your hardware you now have lost your activation!)
    Now in the configuration window you have to change *Firmware*:
    BIOS --> UEFI ovmf-x86_64-ms-4m-code.bin
    This is UEFI with Microsoft key

    Push Button insert new device -> select TPM -> change passthrough to emulate

    Then select: Begin Installation and keep your fingers crossed
    Windows flips in Repair Mode - but should handle this issue.

- We increase disk-size: prepare a bootable gparted USB stick (search the web!) :
  - increase virt-size: `qemu-img resize var/lib/libvirt/images/win11.qcow2 +50G`
    view filesystems: `virt-filesystems -d win11 --filesystems -l`

in my case:
free space was inserted AFTER uefi and windows repair partition - so it is not usable

- now our bootable USB comes into play: plug it in now!
  VM is down: with *virt-manager* we view details of VM now:
  push button *add device* and select your USB-stick
  go to *boot options* and move USB-stick upward to upmost level
  boot now VM into *gparted* USB-stick:
  move all partitions behind your windows main partition (you can determine it if you look at the size) to the end of free space(UEFI-partition, windows-recovery-partition, ….)

  unallocated space should be now immediate behind your windows partition
  now resize windows-main partition (you could do this in windows -
  but why? Be aware that partition numbers did not change!)
  exec the tasks in *gparted* and then quit (shutdown VM)

- remove USB-stick from VM (*virt-manager*) and on the host machine - OK boot now
  verify disk size with disk-manager in windows

- in my case the screen resolution was lousy 800x600 - I changed it with:
  - during boot of VM press F2 to enter UEFI
  - select Device Manager --> OVMF platform configuration
    change resolution (I got best result with 1200x1080) F10 ESC ESC
  - select RESET (label is misleading - uses current config for next boot)

- Windows Activation: Windows Support and convince the guys that you have changed your hardware and therefore lost your activation or
  buy an activation key for € 20.- in the web (what I did)

- Download *WindowsPCHealthCheckSetup.msi* and check if you are Windows11 ready now.
  Wait that Windows Update loads down the Windows 11 upgrade.

Personal remarks:
I searched the web for days(weeks) - all other recommendations (use virtio instead of VGA,
use VNC instead of spice, install network card with virtio driver from Redhat virtio-win.iso)
were all misleading in my case - some configs worked but the performance was horrible, especially screen resolution.
My Network configuration was done by libvirtd with NAT - and it worked out of the box.
You can list the configuration with **iptables -L** (as root)
Booting windows needs only 5 seconds - I can live with that!