Win 11 on Linux KVM/QEMU

Real requirements

- 4 GB RAM (can be shared with Linux).
- 12 GB storage space.
- 64 Bit CPU with 2 cores and with QEMU supported Virtualisation/Hypervisor Features.

Not required

Everything else MS is claiming about requirements at ”Windows 11 Specs and System Requirements“ is not true.

Linux Software

Install QEMU with Virtual-Manager and all of its dependencies.

```
sudo pacman -S qemu-desktop virt-manager iptables-nft init-libvirt dnsmasq
(:: iptables-nft and iptables are in conflict. Remove iptables? [y/N] y)
(Packages =>89 !)
```

I recommend using the **Linux Kernel with the ZEN patches**. It runs better with a virtual host load than a stock kernel.

```
sudo pacman -S linux-zen
```

If you have a lot of RAM, you also want to add "preempt=voluntary" to the GRUB_CMDLINE_LINUX_DEFAULT or GRUB_CMDLINE_LINUX line in /etc/defaults/grub.
Activate your GRUB changes:

```
sudo update-grub
```

Activate the libvirt user mode:
Uncomment (#) the following lines in /etc/libvirt/libvirtd.conf :

```
# UNIX socket access controls
unix_sock_group = "libvirt"
unix_sock_ro_perms = "0777"
unix_sock_rw_perms = "0770"
```

Activate the libvirt daemon with the boot system tool of your choice.
With the OpenRC boot system:

```
sudo rc-update add libvirtd default
```

Add your default user to the libvirt group:

```
sudo usermod -a -G libvirt exampleusername
```

Reboot into the Linux ZEN kernel.

Checks:
- What linux kernel do I run?

```
uname -a
```

- Has my linux kernel command line ‘preempt=voluntary’ in it?

```
cat /proc/cmdline
```

- Is the libvirt daemon running?

```
ps xa | grep libvirt[d]
```

- Is my default user member of the group libvirt?

```
id exampleusername
```

Windows Software

The easiest way to get a Windows 11 Pro installation image is to download one from MS.
You want to '**Download Windows 11 Disk Image (ISO) for x64 devices'** (Windows 11 multi-edition ISO for x64 devices).
"Select the product language" = "**English (United States)**". You can add additional languages later.
-> "**64-bit Download**"
You will get a file named about: **Win11_22H2_English_x64v2.iso** (about 5.3 GB).
This is your Windows 11 Pro installation image. "**Verify your download**" with the `sha256sum` program.

Get the latest virtIO .iso file with the latest **SPICE guest tools and windows side-load drivers**.
Version virtio-win-0.1.240.iso tested.

You probably want to activate your Win11 to be able to test all Windows 11 features.
A very clever guy has found an interesting way to ask MS for a permanent digital license to archive that.
Version 1.4.8 tested. 5512D79C0FC3A0813A9FB9540CFED662C5BA607C

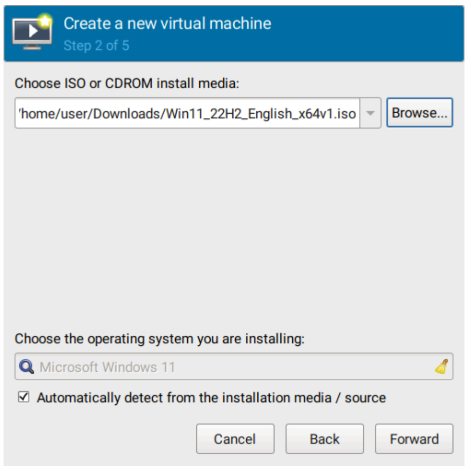
You definitely want to get rid of all the unnecessary Windows 11 crap with BloatyNosyApp. Version 0.85 tested
(If you are a clever guy 👨🏻‍💻 who got the Android Windows Subsystem to work under QEMU, please let me know how.)

There is a promising new script that builds a trimmed-down Windows install .iso from your downloaded original MS Win11 .iso Tiny11Builder. This saves a lot of work.

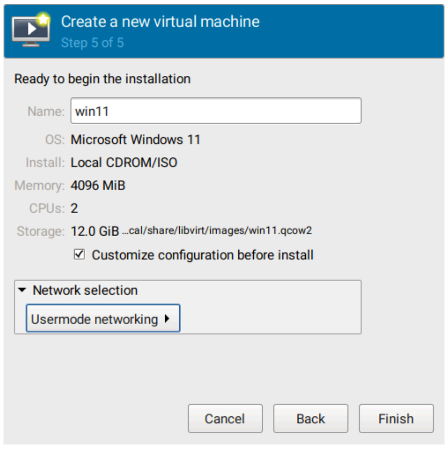
Setting up KVM/QEMU

We want to run QEMU in the way saver **user mode**, and not in the system (root) mode.
Almost all of the defaults of QEMU 7.1 and Virtual-Machine-Manager 4.1 and newer are fine.

Start the Virtual Machine Manager the 1st time from a terminal with:
```
virt-manager -c qemu:///session
```
- Main-menu -> Edit -> Connection Details
  Overview -> Basic details -> Autoconnect: [✔️]
  File -> Close
- Main-menu -> File -> New Virtual Machine
- Choose how you would like to install the operating system: (🔘) Local install media


Create a new virtual machine
Step 2 of 5
Choose ISO or CDROM install media: 'home/user/Downloads/Win11_22H2_English_x64v1.iso
Choose the operating system you are installing: Microsoft Windows 11
☑ Automatically detect from the installation media / source

  Choose the downloaded Windows 11 Pro installation image.
- Choose Memory and CPU settings:
  Memory: 4096 MB minimum.
  CPUs: 2 minimum.
- [✔️] Enable storage for this virtual machine
  12 GB minimum. 65 GB recommended. The default storage directory is: ~/.local/share/libvirt/images/


Create a new virtual machine
Step 5 of 5

Ready to begin the installation

Name: win11
OS: Microsoft Windows 11
Install: Local CDROM/ISO
Memory: 4096 MiB
CPUs: 2
Storage: 12.0 GiB ...cal/share/libvirt/images/win11.qcow2
☑ Customize configuration before install

Network selection
Usermode networking

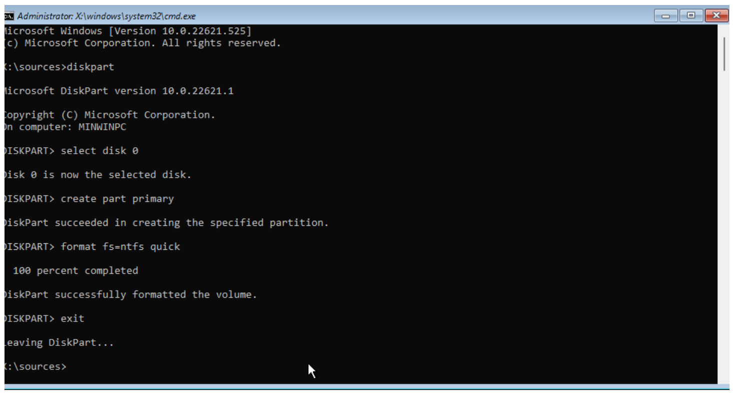
- [✔️] Customize configuration before install -> [Finish]

- **Overview:**
  Hypervisor Details
  Firmware: **BIOS** -> [Apply]
  NIC :xx:xx:xx -> Device model: **virtio** (workaround for the link state QEMU bug)
  Link state: **[ ]** active (set it to <u>not</u> active (is not respected by QEMU)) -> [Apply]
  [**Add Hardware**]
  Storage
  Device type: CDROM device
  (🔘) Select or create custom storage
  Manage… (choose: /home/user/Downloads/virtio-win-0.1.240.iso)
  [Finish]
  -> [Apply]
- **-> Begin Installation**

Windows Install

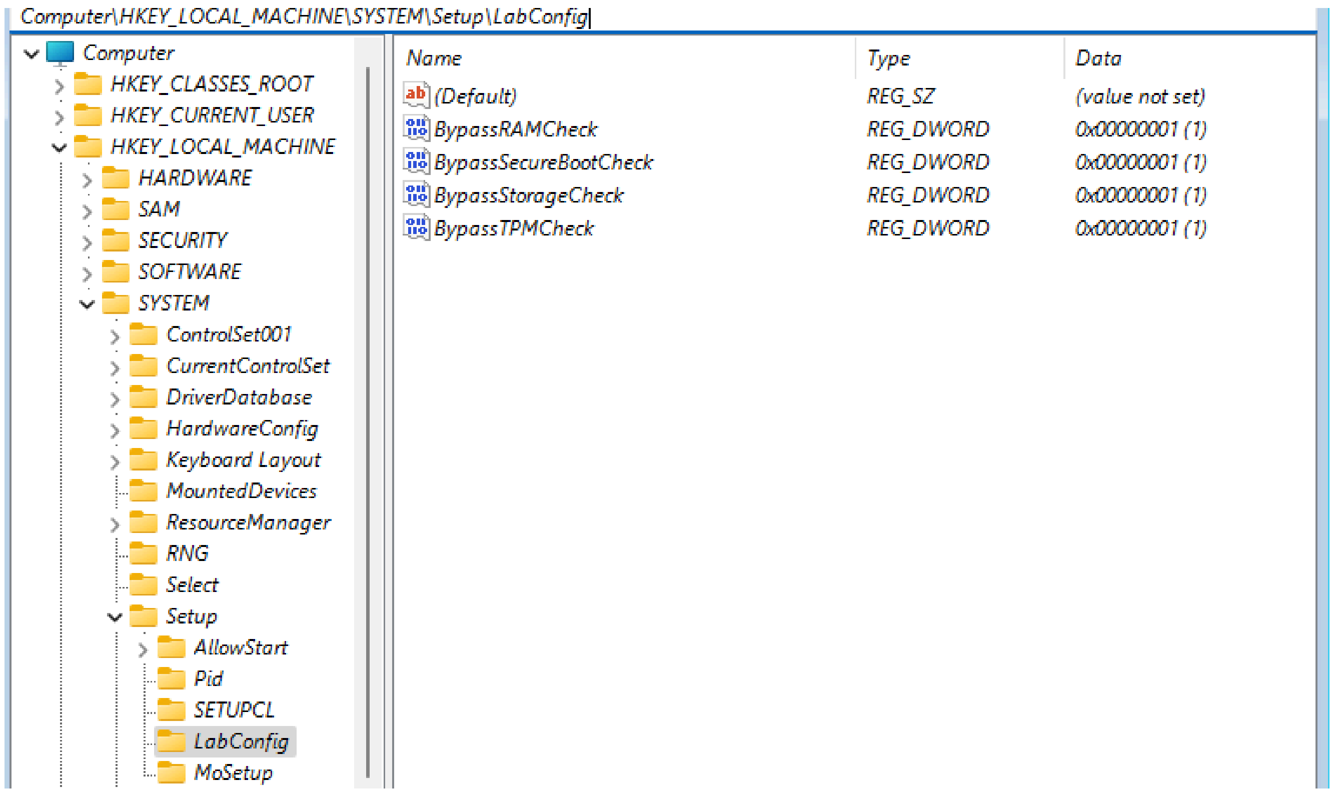
At the 1st ‘Windows Setup’ window press **Shift** + **F10** for a command prompt.

- Install Windows in a single partition in BIOS/MBR mode (F… off with UEFI “Secure” Boot).
  `diskpart`
  DISKPART> `select disk 0`
  DISKPART> `create part primary`
  DISKPART> `format fs=ntfs quick`


Administrator: X:\windows\system32\cmd.exe

```
icrosoft Windows [Version 10.0.22621.525]
c) Microsoft Corporation. All rights reserved.

:\sources>diskpart

icrosoft DiskPart version 10.0.22621.1

opyright (C) Microsoft Corporation.
n computer: MINWINPC

ISKPART> select disk 0

isk 0 is now the selected disk.

ISKPART> create part primary

iskPart succeeded in creating the specified partition.

ISKPART> format fs=ntfs quick

  100 percent completed

iskPart successfully formatted the volume.

ISKPART> exit

eaving DiskPart...

:\sources>
```

  DISKPART> `exit`
- Bypass all artificial and unnecessary Windows Setup checks.
  `regedit`
  Navigate to "Computer\HKEY_LOCAL_MACHINE\\SYSTEM\\Setup"
  Right-click on the "Setup" key and select "New => Key".
  `"LabConfig"`
  Right-click on the "LabConfig" key and select "New => DWORD (32-bit) Value” and create a value named "**BypassTPMCheck**", and set its data to "**1**".
  With the same steps create the "**BypassRAMCheck**", "**BypassSecureBootCheck**" and "**BypassStorageCheck**" values and set the data also to "**1**".
  You may also like to create: Computer\\HKEY_LOCAL_MACHINE\\SYSTEM\\Setup\MoSetup
  And add a DWORD 32 Bit key "**AllowUpgradesWithUnsupportedTPMOrCPU**" = **1**


Computer\HKEY_LOCAL_MACHINE\SYSTEM\Setup\LabConfig

Name	Type	Data
(Default)	REG_SZ	(value not set)
BypassRAMCheck	REG_DWORD	0x00000001 (1)
BypassSecureBootCheck	REG_DWORD	0x00000001 (1)
BypassStorageCheck	REG_DWORD	0x00000001 (1)
BypassTPMCheck	REG_DWORD	0x00000001 (1)

Close the Registry Editor.

```
exit
```

- ‘Windows Setup’ -> [Next] -> [Install now]
- ‘Windows Setup’ -> ‘**I don’t have a product key**’
- ‘Windows Setup’ Select the operating system you want to install -> **Windows 11 Pro** -> [Next]
- ‘Windows Setup’ Custom: **Install Windows only (advanced)**
- ‘Windows Setup’ Where do you want to install Windows? -> **Drive 0 Partition 1** -> [Next]

If something goes wrong… Look in the Windows Setup log file: Press **Shift** + **F10** for a command prompt.

```
more \windows\panther\setuperr.log
```

- Bypass the Out-of-the-Box-Experience (OOBE) program Internet requirement.
  Press **Shift** + **F10** for a command prompt.
  ```
  oobe\bypassnro
  ```
  Windows will restart and the OOBE program starts again. But this time you have an additional choice at “Let’s connect you to a network” -> ‘**I don’t have internet**’ -> ‘**Continue with limited setup**’
- OOBE: Who’s going to use this device? **Keep the user name simple.** Use only US-ASCII, no spaces. How about just “user” ?
  Enter a password: **Don’t**. Just click [Next]
- OOBE: Choose privacy settings for your device: **No, No, No, No, No and No.** -> [Accept]

Install E:\virtio-win-guest-tools.exe (all options).

Windows <-> Linux <-> World communication

First things first, delete the Windows Defender.

It is a useless piece of MS ~~sh..~~ software that only hampers you.
Forget about switching it off in Windows. MS will switch it on again with every Windows update.

- Shut down Windows.
- Mount the QEMU image and delete Windows Defender:

```
    sudo modprobe -v nbd max_part=8
    sudo qemu-nbd -c /dev/nbd0 --fork -t -k /tmp/qsock -f qcow2
~/.local/share/libvirt/images/win11.qcow2
    sudo mount -o rw /dev/nbd0p1 /mnt
```

```
sudo rm -v /mnt/Windows/System32/smartscreen*
sudo rm -vrf /mnt/Program\ Files/Windows\ Defender/*
sudo rm -vrf /mnt/Program\ Files/Windows\ Defender\ Advanced\ Threat\
Protection
```

Copy the Windows tools into the QEMU image.

```
sudo 7z x -o/mnt/Windows/Temp/BNA/ ~/Downloads/BloatyNosyApp.zip
sudo mkdir -pf /mnt/Windows/Temp/WDA
sudo cp -af ~/Downloads/W10\ Digital\ Activation\ 1.4.8/W10\ Digital\
Activation\ 1.4.8/Setup/W10DigitalActivation* /mnt/Windows/Temp/WDA/
```

- Umount the QEMU image:

```
sudo umount /mnt
sudo qemu-nbd -d /dev/nbd0 -k /tmp/qsock
sudo modprobe -rv nbd
```

Take a QEMU snapshot and switch the Internet for Windows on.

Virtual Machine Manager -> QEMU/KVM User sessions
-> Double click on your win11 virtual machine entry to open the virtual machine management window.
View -> 🔘 Snapshots -> [+] **Create new snapshot**
View -> 🔘 Details -> NIC :xx:xx:xx -> Device model: **e1000e** (workaround for the link state QEMU bug)
Link state: **[✔] active** -> [Apply]
View -> 🔘 Console
Virtual Machine -> **Run**

Run your win11 virtual machine and do the Windows de-crapping, activating and updating.

If the win11 “security center” bugs you with notifications, create a file named notifications_off.reg with:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender Security Center\Notifications]
"DisableNotifications"=dword:00000001
```

in it and run it with regedit.

Samba for your virtual machine.

You need `sudo pacman -S samba init-samba smbclient`

- A minimalistic /etc/smb.conf

```
[global]
workgroup = WORKGROUP
server role = standalone server
[tmp]
comment = Temporary file space to exchange data
path = /tmp
read only = no
public = yes
```

- Add your default Linux user to the Samba password database.

```
sudo smbpasswd -a exampleusername
```

- There is no need to run the Samba daemon all the time. You just add an attack surface.
  With this 4 files you are able to convenient switch Samba on and off:
  ~/.local/share/applications/Filesharing_Start.desktop

```
#!/usr/bin/env xdg-open
[Desktop Entry]
Type=Application
Version=1.0
Name=Filesharing start
GenericName=Start the samba filesharing service
Comment=
Icon=gtk-execute
Exec=sudo /usr/local/sbin/samba_start.sh
Terminal=true
Categories=System;Settings;HardwareSettings;
```

~/.local/share/applications/Filesharing_Stop.desktop

```
#!/usr/bin/env xdg-open
[Desktop Entry]
Type=Application
Version=1.0
Name=Filesharing stop
GenericName=Stop the samba filesharing service
Comment=
Icon=gtk-close
Exec=sudo /usr/local/sbin/samba_stop.sh
Terminal=true
Categories=System;Settings;HardwareSettings;
```

/usr/local/sbin/samba_start.sh

```
#!/bin/bash
/etc/init.d/smb restart ; /etc/init.d/smb status
sleep 3
```

/usr/local/sbin/samba_stop.sh

```
#!/bin/bash
/etc/init.d/smb stop ; /etc/init.d/smb status
sleep 3
```

Test Samba: `smbclient //localhost/tmp`

Arch Linux has a Wiki entry about KVM/QEMU/libvirt https://wiki.archlinux.org/title/Libvirt.
But I think this Wiki entry is too complicated for the average user and does not cover Windows at all.

If there is interest, I will add how to use https://github.com/crazy-max/WindowsSpyBlocker/tree/master/data/hosts with libvirt.