# Macos VM on Windows

[https://github.com/myspaghetti/macos-virtualbox/blob/master/macos-guest-virtualbox.sh](https://emea01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fmyspaghetti%2Fmacos-virtualbox%2Fblob%2Fmaster%2Fmacos-guest-virtualbox.sh&data=04%7C01%7C%7C98cee942dd4f45e9098b08d9c124f692%7C84df9e7fe9f640afb435aaaaaaaaaaaa%7C1%7C0%7C637753186757140138%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=EasfY0ANUWKLiWYtOI4vKkgNq3rKNtZKph5LJiRklAQ%3D&reserved=0)

# Macos VM on Linux

[https://github.com/foxlet/macOS-Simple-KVM](https://emea01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Ffoxlet%2FmacOS-Simple-KVM&data=04%7C01%7C%7C06a5f5dcf82e4d6f7d9a08d9e8fadd1e%7C84df9e7fe9f640afb435aaaaaaaaaaaa%7C1%7C0%7C637796986406167840%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=lvtvTFWWRajuHb7mSuTcMDQOpFqbNMZob7lQUtgFVmA%3D&reserved=0) (alternative 1 : with Clover)

[https://github.com/kholia/OSX-KVM](https://emea01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkholia%2FOSX-KVM&data=04%7C01%7C%7C1ac827bedcf444679f2208d9e8fb44f9%7C84df9e7fe9f640afb435aaaaaaaaaaaa%7C1%7C0%7C637796988149797698%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=HXlGDahvTzBZxmioiAz51KRO%2FPKSzgduNVUFItXWxPM%3D&reserved=0) (alternative 2 : with OpenCore)

[Virtualiser MacOS sur Linux avec QEMU – Tuxicoman (jesuislibre.net)](https://tuxicoman.jesuislibre.net/2021/01/virtualiser-macos-sur-linux-avec-qemu.html)

Je vous conseille les scripts de [foxlet/macOS-Simple-KVM](https://github.com/foxlet/macOS-Simple-KVM)

Voici ma config perso (corrige le problème de pointeur de souris, meilleure performance i/o & graphique, exposition par le protocole SPICE) :

#!/bin/bash OSK="ourhardworkbythesewordsguardedpleasedontsteal(c)AppleComputerInc" VMDIR=$PWD OVMF=$VMDIR/firmware [#export](https://tuxicoman.jesuislibre.net/tag/export) QEMU\_AUDIO\_DRV=pa [#QEMU\_AUDIO\_DRV](https://tuxicoman.jesuislibre.net/tag/qemu_audio_drv)=pa qemu-system-x86\_64 \ -nodefaults \ -enable-kvm \ -m 6G \ -machine q35,accel=kvm \ -smp 4,cores=2 \ -cpu Penryn,vendor=GenuineIntel,kvm=on,+sse3,+sse4.2,+aes,+xsave,+avx,+xsaveopt,+xsavec,+xgetbv1,+avx2,+bmi2,+smep,+bmi1,+fma,+movbe,+invtsc \ -device isa-applesmc,osk="$OSK" \ -smbios type=2 \ -drive if=pflash,format=raw,readonly,file="$OVMF/OVMF\_CODE.fd" \ -drive if=pflash,format=raw,file="$OVMF/OVMF\_VARS-1024x768.fd" \ -vga virtio \ -device ich9-intel-hda -device hda-output \ -usb -device usb-kbd -device usb-tablet \ -netdev user,id=net0 \ -device vmxnet3,netdev=net0,id=net0,mac=52:54:00:09:49:17 \ -drive id=ESP,if=virtio,format=qcow2,file=ESP.qcow2 \ -drive id=MyDisk,if=virtio,format=qcow2,file=MyDisk.qcow2 \ -k fr \ -spice port=3001,disable-ticketing \ -device virtio-serial -chardev spicevmc,id=vdagent,debug=0,name=vdagent \ -device virtserialport,chardev=vdagent,name=com.redhat.spice.0 \

Ca se regarde avec spicy:

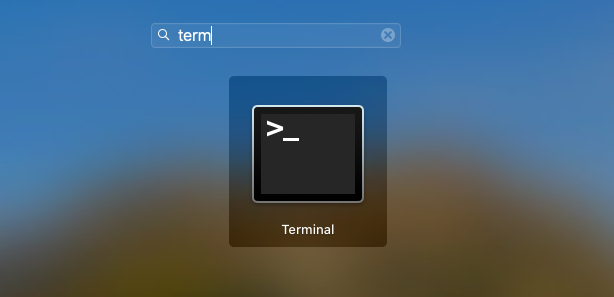
spicy --uri=spice://localhost:3001

# Quemu install on Macos Catalina

1. **Open Terminal**



click Launchpad (rocket icon)



input "term"

click on Terminal

# Une image contenant texte Description générée automatiquement

1. **Install Howebrew**

Une image contenant texte

Description générée automatiquement

Before … (disk usage : 7396 MB)

Une image contenant texte

Description générée automatiquement

After … (disk usage : 9279 MB => install size = 1883 MB)

Install time : 10 min

* Sudo password
* RETURN to confirm install plan
* (downloading & installing command line tools for XCode takes time : 5 min)
* (downloading & installing homebrew/core takes time : 460 MB – 5 min)

[https://brew.sh](https://emea01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fbrew.sh%2F&data=04%7C01%7C%7Cdac3339da77641db580108d9e8bff7e8%7C84df9e7fe9f640afb435aaaaaaaaaaaa%7C1%7C0%7C637796733463195492%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=t9FLGYBeFQ9CURogAn5qn2bNjlQVDAZkobex6ftmmuU%3D&reserved=0)

> /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

Une image contenant texte

Description générée automatiquement

Interaction 1 : enter sudo password to execute the install script

Une image contenant texte

Description générée automatiquement

Interaction 2 : accept the script modifications summary

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Interaction 3 : enter sudo password for xcode-select

Une image contenant texte

Description générée automatiquement

…………

Une image contenant texte

Description générée automatiquement

1. **Install Qemu**



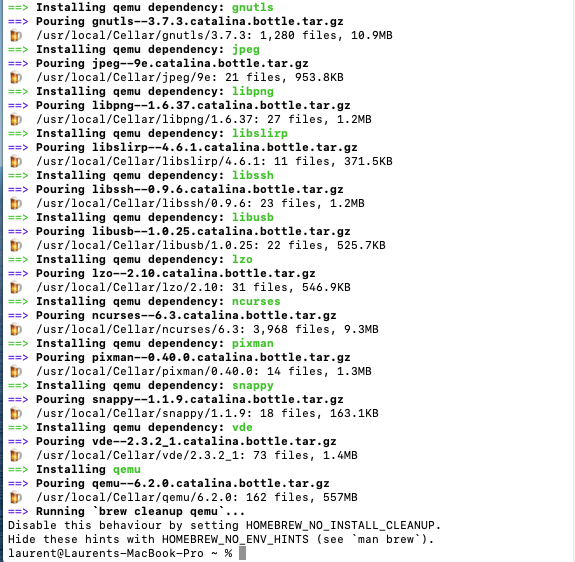
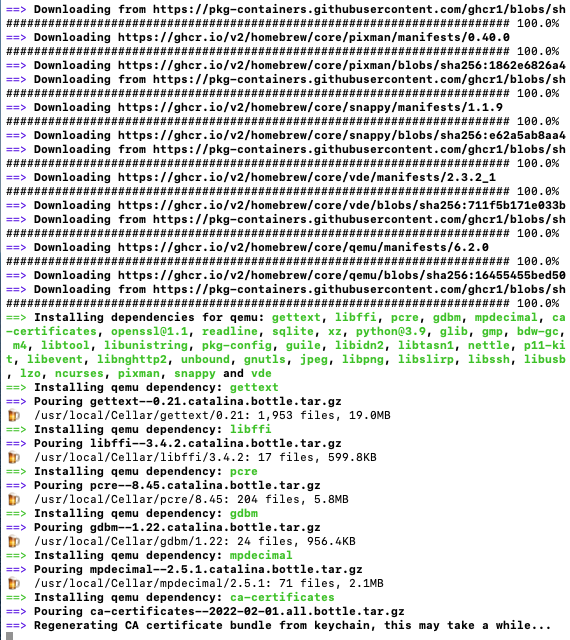
After …

Install time : 5 min – disk space : 1220 MB

[https://www.qemu.org/download/#macos](https://emea01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.qemu.org%2Fdownload%2F%23macos&data=04%7C01%7C%7Cdac3339da77641db580108d9e8bff7e8%7C84df9e7fe9f640afb435aaaaaaaaaaaa%7C1%7C0%7C637796733463195492%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=rjyqm%2FzM36e7uo2Xl7uVUrLxyroyGjrXRmpBGyZQ1zw%3D&reserved=0)

> brew install qemu





# Example using Qemu with Apple Hypervisor Framework on Macos

[Check Hyper-V (Intel VT-x) Virtualization Support On MacOS Computer | Build5Nines](https://build5nines.com/check-hyper-v-intel-vt-x-virtualization-support-on-macos-computer/)

sysctl -a | grep machdep.cpu.features

 look for the VMX feature in the list.

Check if hypervisor is supported :

Supported hardware

The Hypervisor framework requires hardware support to virtualize hardware resources. On Apple silicon, that includes the Virtualization Extensions. On Intel-based Mac computers, the framework supports machines with an Intel VT-x feature set that includes Extended Page Tables (EPT) and Unrestricted Mode.

At runtime, determine whether the Hypervisor APIs are available on a particular machine with the [sysctl](https://developer.apple.com/documentation/kernel/sys" \l "3571071) command, passing kern.hv\_support as an argument.

Entitlements

All process must have the [com.apple.security.hypervisor](https://developer.apple.com/documentation/bundleresources/entitlements/com_apple_security_hypervisor) entitlement to use Hypervisor API.

> sysctl kern.hv\_support

0



Checking nested virtualization in Ubuntu KVM host :

[Nested KVM / QEMU / libvirt Virtualization on Ubuntu (graphica.com.au)](https://tips.graphica.com.au/nested-kvm/)

[How to enable nested virtualization in KVM :: Fedora Docs (fedoraproject.org)](https://docs.fedoraproject.org/en-US/quick-docs/using-nested-virtualization-in-kvm/)

cat /sys/module/kvm\_intel/parameters/nested

Y

To enable nested virtualization for Intel processors:

1. Shut down all running VMs and unload the kvm\_probe module:

sudo modprobe -r kvm\_intel

1. Activate the nesting feature:

sudo modprobe kvm\_intel nested=1

1. Nested virtualization is enabled until the host is rebooted. To enable it permanently, add the following line to the /etc/modprobe.d/kvm.conf file:

options kvm\_intel nested=1

1. Start the virtual machine.
2. On the virtual machine, run:

sudo dnf group install virtualization

1. Verify that the virtual machine has virtualization correctly set up:

sudo virt-host-validate