QEMU

QEMU is a generic and open source machine emulator and virtualizer a.k.a. **hypervisor**. QEMU works like VMWare, VirtualBox and etc.

QEMU is a kind of **hypervisor** running on the host machine which can install various OS as virtual machine. Before reviewing below, you'd better understand Difference between Host and Guest Operating System.

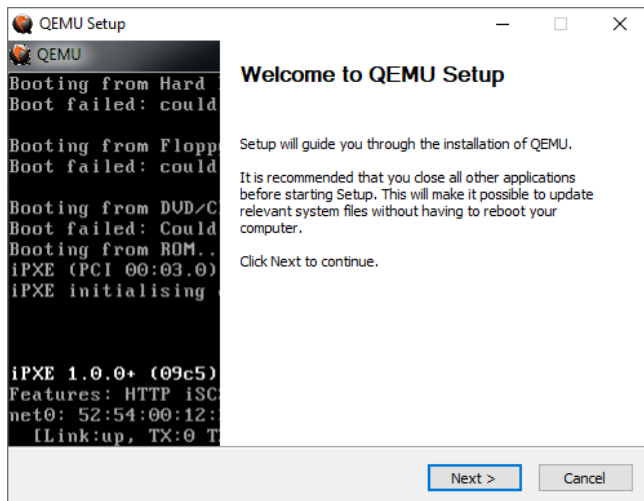- Setup QEMU on Windows 10
- QEMU Command Line Example to Create Windows VM on WIndows 10
- QEMU Command Line Example to Create CentOS 8 VM on WIndows 10
- Setup QEMU on Mac
- QEMU Command Line Example to Create Windows 10 VM on Mac
- Run QCOW2 image without GUI
- Run QCOW2 image with no GUI, forwarding ports (8080 to 80), (20022 to 22)

Setup QEMU on Windows 10
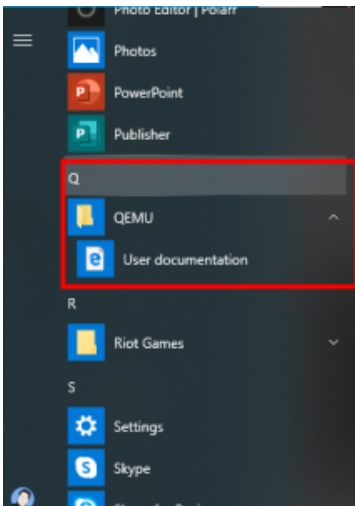
You can download its installation file at https://www.qemu.org/download/

On Windows 10, you will see following diagram when you install QEMU.

You will surprisingly find nothing on the program group except a link to the documentation.

QEMU does not provide VM manager like VMWare, VirtualBox and etc - meaning it requires to run the necessary things like command line interface.

QEMU Command Line Example to Create Windows VM on WIndows 10

Below is an example to create an image by command line interface.

```
"C:\Program Files\qemu\qemu-img.exe" create windows.raw 20G
```

If you want to create in qcow2 with 2048G disk space, you can create it as following:

```
"C:\Program Files\qemu\qemu-img.exe" create -f qcow2 centos7.qcow2 2048G
```

You will need to install operating system, and below command line example shows how to install it by CD-ROM driver - it is actually linked to iso image you have.

```
"C:\Program Files\qemu\qemu-system-x86_64.exe" -m 4G -hda centos7.qcow2 -smp sockets=1,cores=2,threads=2 -cdrom
Win10_1903_V1_Korean_x64.iso
```

Once installation is done, you may not need CD-ROM driver any more, so you can simply run your virtual machine like below - it also shows how to use audio device.

```
"C:\Program Files\qemu\qemu-system-x86_64.exe" -m 4G -hda centos7.qcow2 -enable-kvm -cpu host -smp sockets=1,
cores=2,threads=2 -drive file=windows.raw,format=raw -soundhw all
```

QEMU Command Line Example to Create CentOS 8 VM on WIndows 10

Below is an another example to create CentOS8 image

```
"C:\Program Files\qemu\qemu-img.exe" create centos8.raw 100G
```

Setup QEMU on Mac

```
brew install qemu
```

QEMU Command Line Example to Create Windows 10 VM on Mac

```
qemu-img create -f qcow2 windows10.qcow2 512G
```

Below is for OS installation

```
qemu-system-x86_64 \
        -m 4G \
        -vga virtio \
        -display default,show-cursor=on \
        -usb \
        -device usb-tablet \
        -machine type=q35,accel=hvf \
        -smp 2 \
        -cdrom ~/Downloads/windows_10.iso \
        -hda ~/Documents/windows10.qcow2
```

Below is to run windows 10 VM without cdrom

win.sh

```
#!/bin/bash
qemu-system-x86_64 \
        -m 4G \
        -vga virtio \
        -display default,show-cursor=on \
        -usb \
        -device usb-tablet \
        -machine type=q35,accel=hvf \
        -smp 2 \
        -hda ~/Documents/windows10.qcow2
```

If you want to use full screen, just add -full-screen in the parameter.

To run windows in the background without displaying anything on the terminal:

win.sh

```
#!/bin/bash
pkill qemu
qemu-system-x86_64 \
        -m 8G \
        -vga virtio \
        -usb \
        -device usb-tablet \
        -machine type=q35,accel=hvf \
        -smp 4 \
        -hda ~/Documents/windows10.qcow2 \
        -device e1000,netdev=user.0  \
        -netdev user,id=user.0,hostfwd=tcp::3389-:3389 \
        -display none &
```

Run QCOW2 image without GUI

Below example shows how to run QCOW2 image without GUI on linux terminal

```
$ qemu-system-x86_64 -cdrom /home/zyh/ubuntu-16.04.3-server-amd64.iso -hda Ubuntu16.04.qcow2 -boot d -net nic -
net user -m 1024 -localtime -nographic
```

Run QCOW2 image with no GUI, forwarding ports (8080 to 80), (20022 to 22)

```
$ qemu-system-x86_64 -m 8G -smp cores=2,threads=2 -hda centos8.qcow2 -device e1000,netdev=user.0 -netdev user,
id=user.0,hostfwd=tcp::80-:80,hostfwd=tcp::20022-:22 -display none
```

If you run your virtual machine above, you will be able to connect to your server via ssh by **ssh 127.0.0.1 -p 20022**, and you can connect to your web server in VM by **lynx http://127.0.0.1:8080**