



How To Install And Confi

Ubuntu

By **SK**



About QEMU

As a Linux enth



QEMU is a generic and open source machine emulator and virtualizer. It is a free and open source tool that allows users to create virtual machines inside the host operating system. The resources of the host, such as Hard drive, RAM, Processor, will be divided and shared among the virtual machines (Virtual machines).

When used as a machine emulator, QEMU can run and emulates machine (e.g. an ARM board) on a different architecture using dynamic translation, it achieves very good performance.

When used as a virtualizer, QEMU achieves near-native performance by executing the guest code directly on the host CPU. QEMU can also run on top of Xen hypervisor or using the [KVM kernel module in Linux](#). When using KVM, QEMU can virtualize x86, server and embedded PowerPC, and S390 guests.

er. It is a free and open source tool that allows users to create virtual machines inside the host operating system, such as Hard drive, RAM, Processor, will be divided and shared among the virtual machines (Virtual machines).

ams made for one architecture (e.g. an ARM board) on a different architecture using dynamic translation, it achieves very good performance.

nces by executing the guest code directly on the host CPU. QEMU can also run on top of Xen hypervisor or using the [KVM kernel module in Linux](#). When using KVM, QEMU can virtualize x86, server and embedded PowerPC, and S390 guests.

QEMU has two operating modes:

- Full system emulation: In this mode, QEMU emulates a full system (for example a PC), including a processor and various peripherals. It can be used to launch different Operating Systems without rebooting the PC or to debug system code.
- User mode emulation (Linux host only): In this mode, QEMU can launch Linux processes compiled for one CPU on another CPU. For example, it can be used to launch Wine or to ease cross-compilation and cross-debugging.

QEMU can run without a host kernel driver and yet gives acceptable performance.

QEMU will work fine in the following hardware:



- Mac99 PowerMac (PowerPC processor, in
- Sun4m/Sun4c/Sun4d (32-bit Sparc proces
- Sun4u/Sun4v (64-bit Sparc processor, in
- Malta board (32-bit and 64-bit MIPS proce
- MIPS Magnum (64-bit MIPS processor) ;
- ARM Integrator/CP (ARM) ;
- ARM Versatile baseboard (ARM) ;
- ARM RealView Emulation/Platform basebo
- Spitz, Akita, Borzoi, Terrier and Tosa PDAs
- Luminary Micro LM3S811EVB (ARM Corte
- Luminary Micro LM3S6965EVB (ARM Cort
- Freescale MCF5208EVB (ColdFire V2) ;
- Arnewsh MCF5206 evaluation board (ColdFire V2) ;
- Palm Tungsten|E PDA (OMAP310 processor) ;
- N800 and N810 tablets (OMAP2420 processor) ;
- MusicPal (MV88W8618 ARM processor) ;
- Gumstix "Connex" and "Verdex" motherboards (PXA255/270) ;
- Siemens SX1 smartphone (OMAP310 processor) ;
- AXIS-Devboard88 (CRISv32 ETRAX-FS) ;
- Petalogix Spartan 3aDSP1800 MMU ref design (MicroBlaze) ;
- Avnet LX60/LX110/LX200 boards (Xtensa).

For user emulation, x86 (32 and 64 bit), PowerPC (32 and 64 bit), ARM, MIPS (32 bit only), Sparc (32 and 64 bit), Alpha, ColdFire(m68k), CRISv32 and MicroBlaze CPUs are supported.

1. Install QE

In this let us see h
create and manag



2. Creating Virtual Machine

Now, let us create a new virtual machine. For this tutorial, let us create a new Ubuntu 15.04 server.

First create the Virtual hard disk image for 20GB size hard disk image.

```
qemu-img create ubuntu.img 20G
```

Or you can create the image with Qemu's native 'qcow2' using the following command:

```
qemu-img create -f qcow2 ubuntu.qcow 20G
```

Then, download the [Ubuntu 15.04 server installation](#) image and boot the Virtual machine. I already downloaded the installation image and kept it in my home directory.

To boot the Virtual machine, run:

```
qemu-system-x86_64 -hda ubuntu.img -boot d -cdrom /home/sk  
/Soft_Backup/OS\ Images/New/ubuntu-15.04-server-amd64.iso -m 640
```

For 32bit VM, use:

```
qemu-system-i386 -hda ubuntu.img -boot d -cdrom /home/sk  
/Soft_Backup/OS\ Images/New/ubuntu-15.04-server-i386.iso -m 640
```

Here, I have allocated







```
^ sudo -i
```

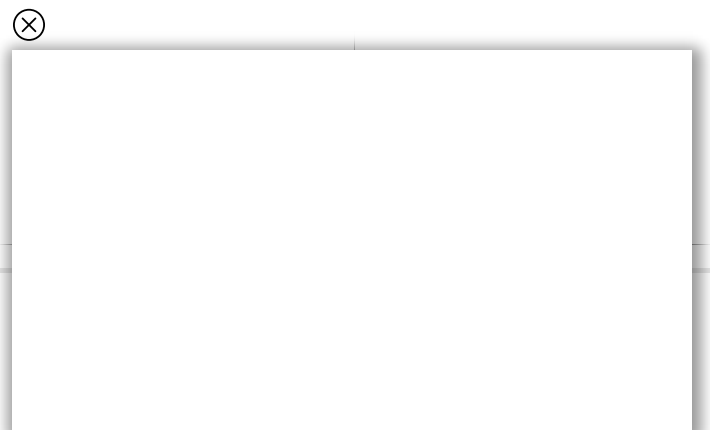
That's it.

Don't like command line way? Creating an VM might be bit difficult to some user. No worries, there is a GUI tool called [Virtual Machine Manager](#) (Virt Manager).

the command line
/Ms using GUI tool

3. Create and manage Virtual Machines using Virt Manager

Launch **Virtual Machine Manager**(Virt Manager) from the Menu or Unity Dash. The default Interface of Virt Manager



^ Create a virtual machine, go to **File -> New**

Select how would like to install the operating system

You can install guest OS using local ISO image or by importing existing disk images. Also, you can select the Advanced options.

or install from network, PXE server, or select the architecture from



Select the local installation





Choose the amount of RAM and no of CPUs.



Enter your virtual h

ult is 8GB.

A large, empty rectangular text input box with a thin gray border and a subtle drop shadow.



Enter your guest OS name. And, Select the
Finally, click Finish.

Advanced options.



Once you click the



That's it. Now, continue and install the Gues

in normal system.



To view the number of installed guest OSs, go to **File -> View manager**.





To view the Virtual machine details, Select **Virtual Machine Details** and go to **Edit ->**



Click on the 'Show virtual hardware details' from the menu bar.





That's it. Cheers!!

Reference:

- [QEMU Website](#)

MY LATEST VIDEOS



