KVM Virtualization

About This Course

KVM Virtualization

What You Need

What you need to follow along

- Linux Academy Lab servers cannot have KVM installed
- Host testing machine with a minimum of 6GB of memory
- AMD or Intel processor with virtualization enabled
- You can use Fedora 22+, CentOS 7, or a RHEL 7 host

KVM Virtualization

What is KVM

KVM

- KVM is a hypervisor built into the Linux Kernel
- Allows Linux desktops or servers to simulate multiple pieces of of hardware
- KVM uses the QEMU virtual machine format

What is the Difference Between QEMU and KVM?

- QEMU and KVM are two separate software projects
- QEMU is primarily a hardware emulator
- KVM is a kernel module that is used to expose hardware virtualization technologies such as:
  - Intel VT-x or AMD SVM
  - KVM then uses QEMU for the device emulation
- When QEMU and KVM work together, KVM arbitrates access to the CPU and memory, while QEMU emulates hardware resources like hard disks, video cards, USBs and more.

KVM Virtualization

Virtualization

Hardware Virtualization

- Full virtualization
  - Complete simulation of the actual hardware to allow software, which typically consists of a guest machine or virtual machine
  - KVM uses full virtualization

- Para-virtualization
  - The hardware environment is not simulated

Desktop Virtualization

- Desktop virtualization is the concept of separating the logical desktop from the physical machine
- VDI Virtual Desktop Infrastructure

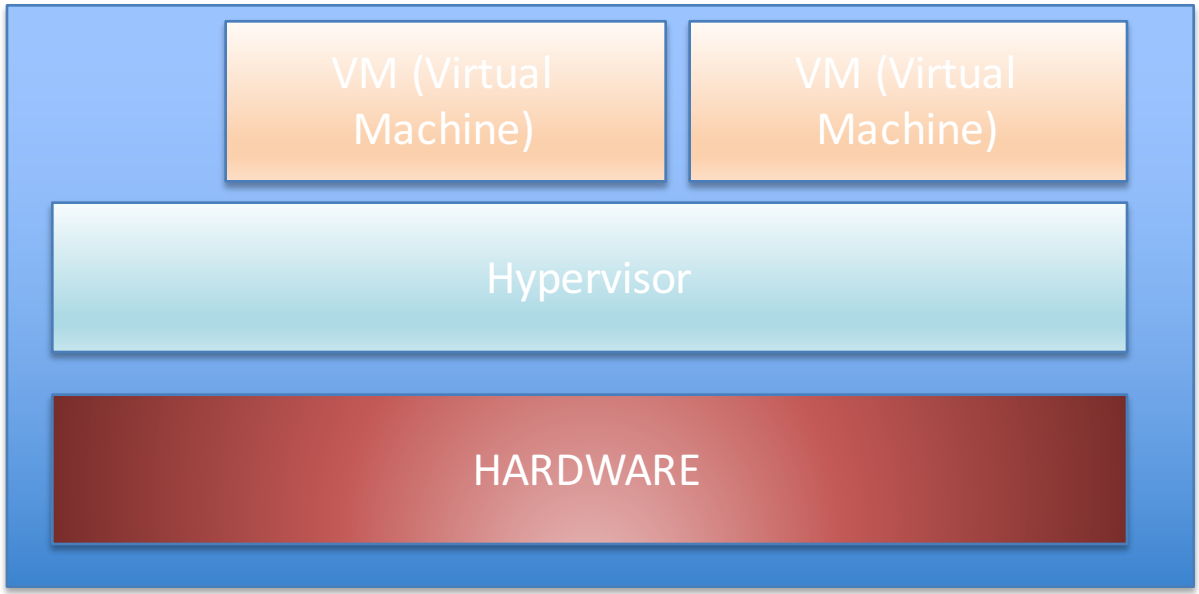
Hypervisors

- VMM/Hypervisor is a piece of software that is responsible for monitoring and controlling virtual machines or guest operating systems

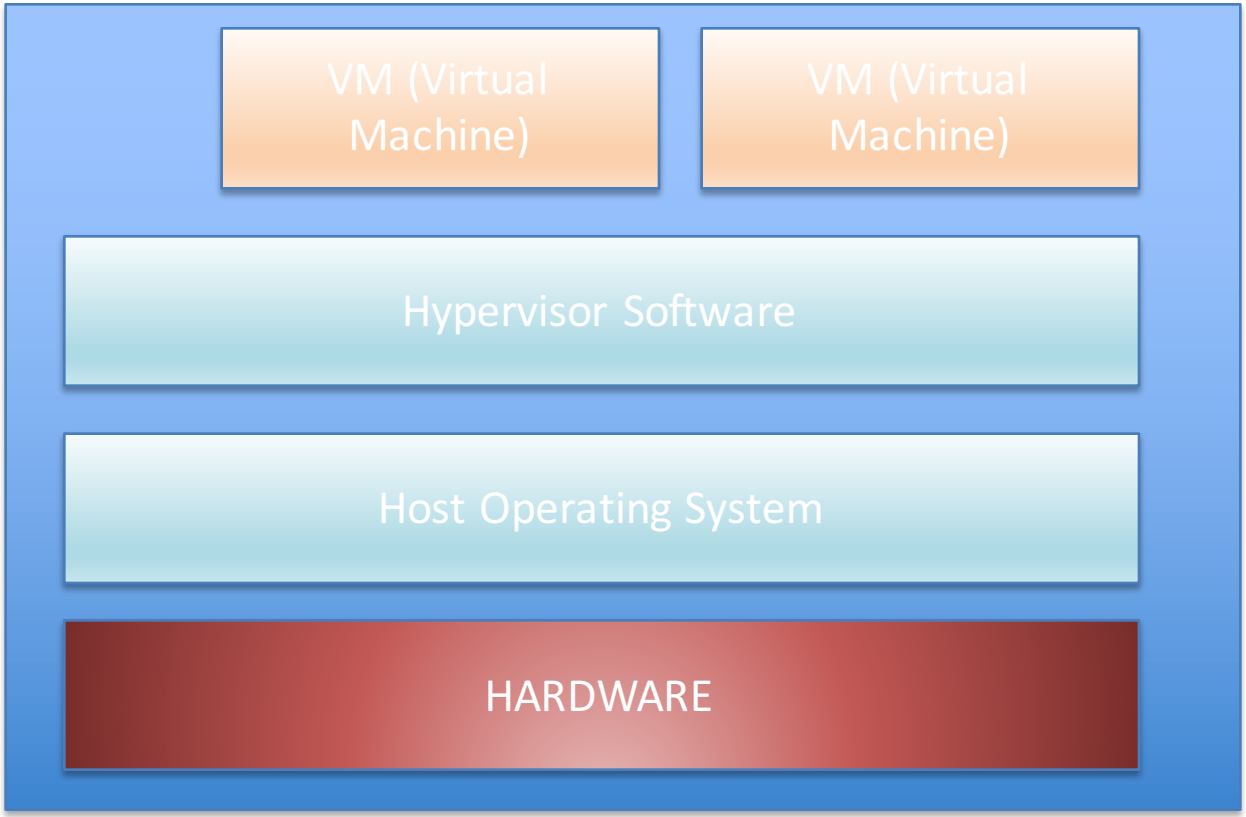
- Type 1 Hypervisor
  - The Hypervisor runs directly on top of the hardware

- Type 2 Hypervisor
  - The Hypervisor acts as a separate layer often on top of a base Operating System

Type 1 Hypervisor



Type 2 Hypervisor



Overcommitting

- To allocate more virtualized CPUs or virtual memory than the available resources on the host system provides
- Overcommitting can cause possible risk to your host system's stability

Thin Provisioning

- Allows you to optimize available storage space for the guest virtual machines
- Similar to overcommitting, but only pertains to storage, not CPU and memory
- Can also pose risk to the system stability

KVM Virtualization

Linux Virtualization

Linux Virtualization

- Process through which one or more virtual machines can be installed, executed and maintained on top of the Linux operating system
- Linux Virtualization brings openness, flexibility, and high performance

Open Source Virtualization projects

- KVM - Kernel-based Virtual Machine
- Xen
- VirtualBox
- UML - User Mode Linux

Kernel Same-Page Merging (KSM)

- Allows KVM guests to share identical memory pages
- Shared common libraries or other identical, high-use data
- KSM allows for greater guest density of identical or similar guest operating systems by avoiding memory duplication

QEMU Guest Agent

- Runs on the guest virtual machine's operating system in order to issue commands to the guest OS from the host OS

Nested Virtualization

- For example, it is possible so that the KVM guest can operate as virtual hosts, essentially allowing users to create one or more KVM guests within each KVM guest

Linux Virtualization and the cloud

- OpenStack
- Eucalytus
- Cloudstack

KVM Virtualization

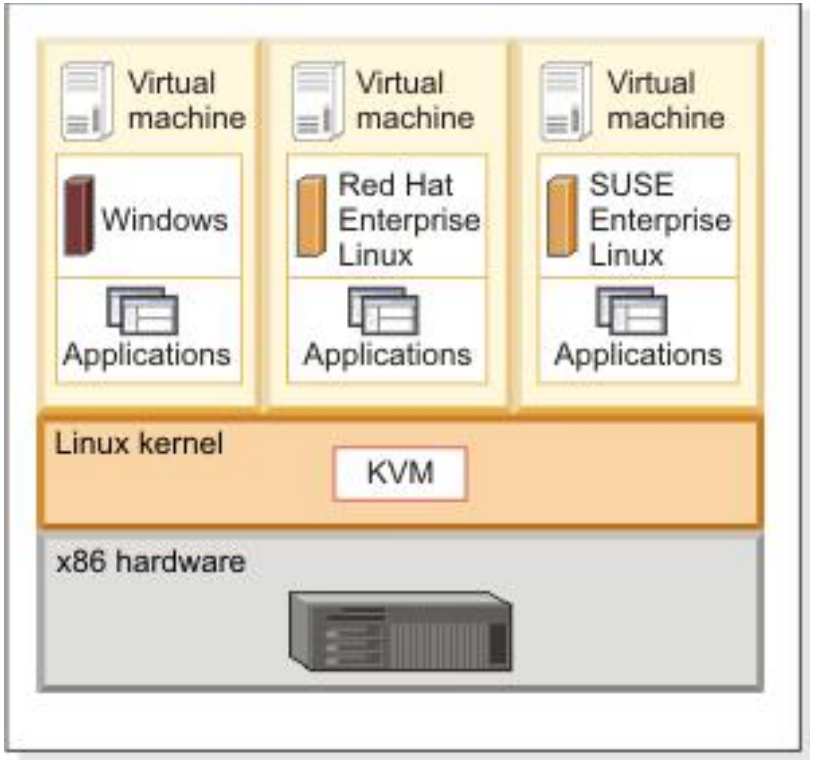
KVM Overview

What is KVM
Kernel-based Virtual Machine

- KVM is a hypervisor which is built into the Linux Kernel
- Allows Linux desktops or servers to simulate multiple pieces of hardware
- Full virtualization solution for Linux on x86 hardware that contains virtualization extensions such as Intel VT or AMD-V
- KVM uses the QEMU (Quick Emulator) virtual machine format

KVM overview

- The KVM module creates a bare metal hypervisor on the Linux Kernel
- Virtual machines can be loaded onto this hypervisor, running separate OSes
- Any storage that is supported by Linux can store virtual machine images (local disks, SCSI or networking-attached storage such as NFS and SAN)

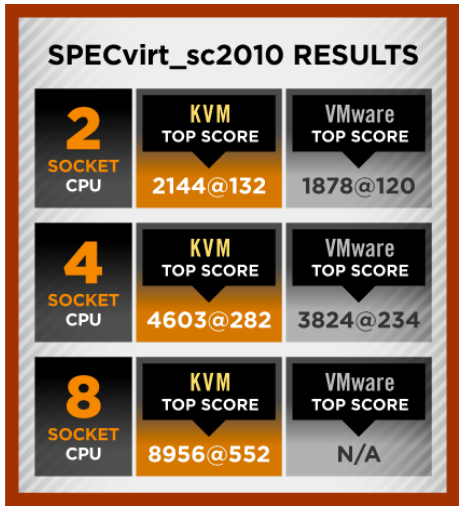


KVM overview

- KVM supports a variety of operating system guest such as:
  - Linux distributions, Microsoft Windows, OpenBSD, FreeBSD, Open Solaris, MS DOS and more

KVM Benefits of Use

- Lower cost
- Enterprise performance and higher scalability
- Advanced security
- High Quality of Service (QoS)
- The open ecosystem



KVM Use Cases

- Hypervisor of choice for cloud
- Enterprise virtualization for large-scale enterprises
- Virtualizing Linux servers
- Virtualization of compute, storage, networking

OpenStack and KVM

- OpenStack is a cloud platform that uses open source technology for both private and public clouds
- KVM is the hypervisor of choice for OpenStack deployments

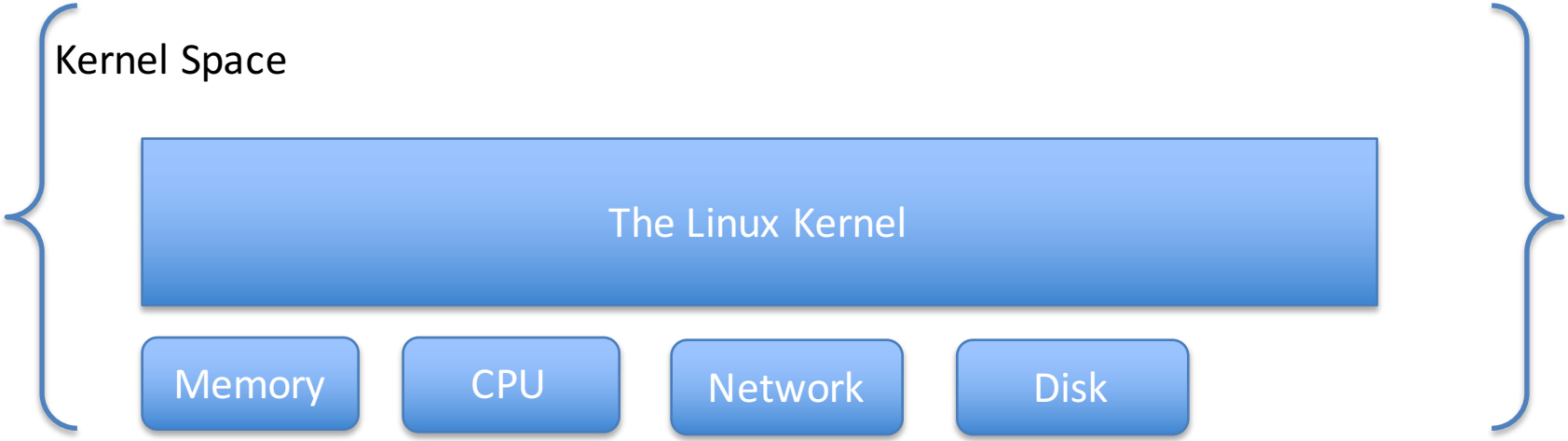
KVM Virtualization

User Space and Kernel Space

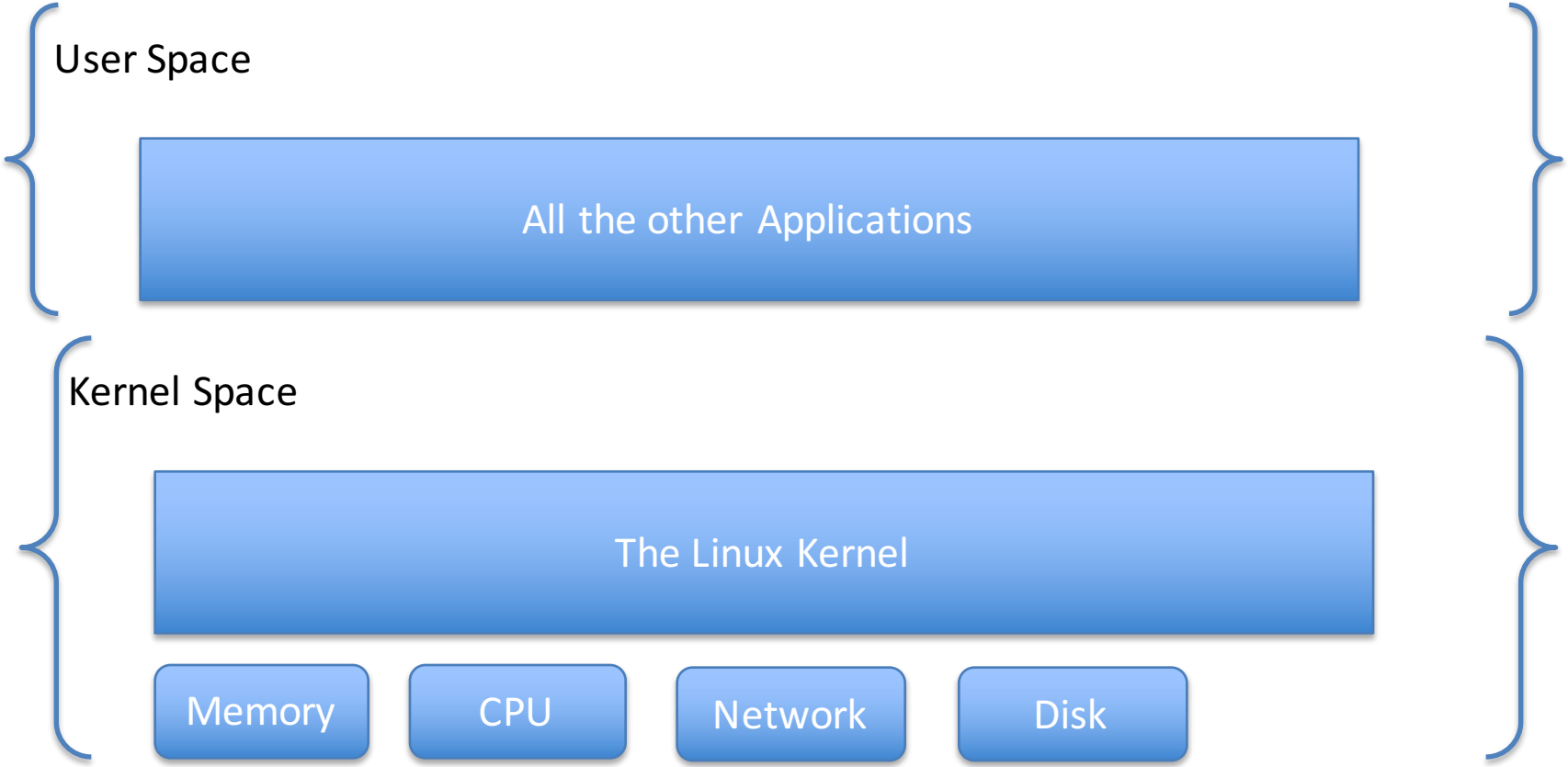
KVM and Linux

- KVM allows us to turn the Linux kernel into a hypervisor
- QEMU is used for I/O emulations with userland (user space) software
  - Userland is user space software that does not use the Linux kernel
  - Computer memory is managed and divided into kernel space and user space

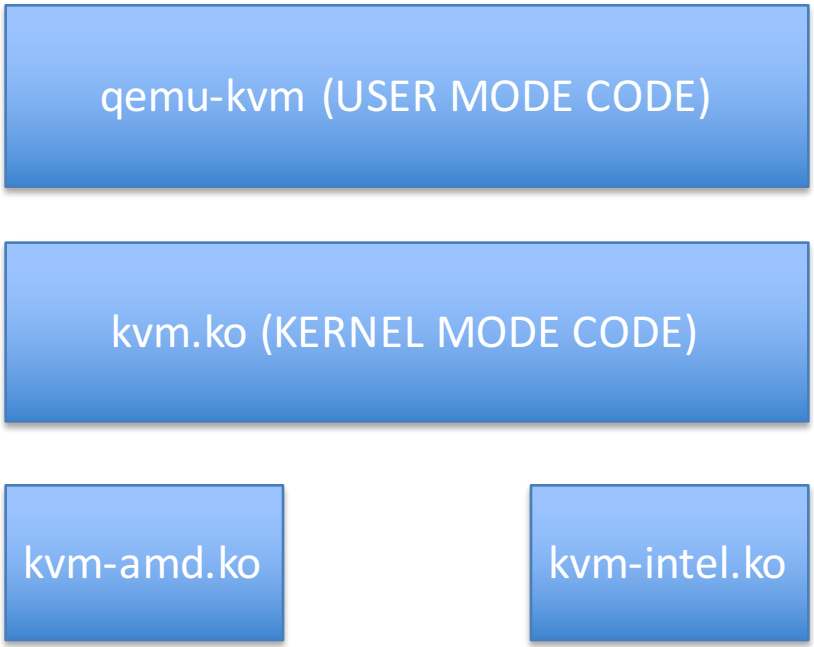
User Space and Kernel Space



User Space and Kernel Space



KVM using user mode and kernel mode



KVM Virtualization

Standalone KVM Virtualization

Virtual Machine Monitor - VMM/Hypervisor

- VMM (Hypervisor)
- Compute software, firmware, or hardware that creates and runs virtual machines
- The KVM kernel Module is not enough to just start running virtual machines…

Quick Emulator - QEMU

- …In order to run these virtual machines KVM also needs an emulator to emaulate the hardware peripherals for the virtual machines
- Open source machine emulator that helps you to run the operating systems that are designed to run one architecture on top of another
- QEMU uses dynamic translation, which is a technique used to execute virtual machine instructions on the host machine

KVM Quick Emulator - KVM QEMU

- QEMU in general is slow
- KVM developers created qemu-kvm, which can interact with KVM modules directly and safely execute instructions from the VM directly to the CPU without using dynamic translations

Libvirt

- Libvirt is a set of API libraries that sites in between the end user and the hypervisor
- The hypervisor could be built to use any virtualization technology such as KVM/QEMU, XEN, LXC, VirtualBox, VMWARE ESX, MS HyperV and even Parallels
- Libvirt acts as a sort of transparent layer that can take commands from users, modifies them based on the underlying virtualization technology, and then executes them on the actual hypervisor
- Tools include the libvirtd daemon, API library, and command line utility called virsh

Libvirt-based tools
virsh

- A command line tool to manage VM Guests with similar functionality as the Virtual Machine Manager. Allows you to change a VM Guest's status (start, stop, pause, etc.), to set up new guests and devices, or to edit existing configurations. virsh is also useful to script VM Guest management operations.

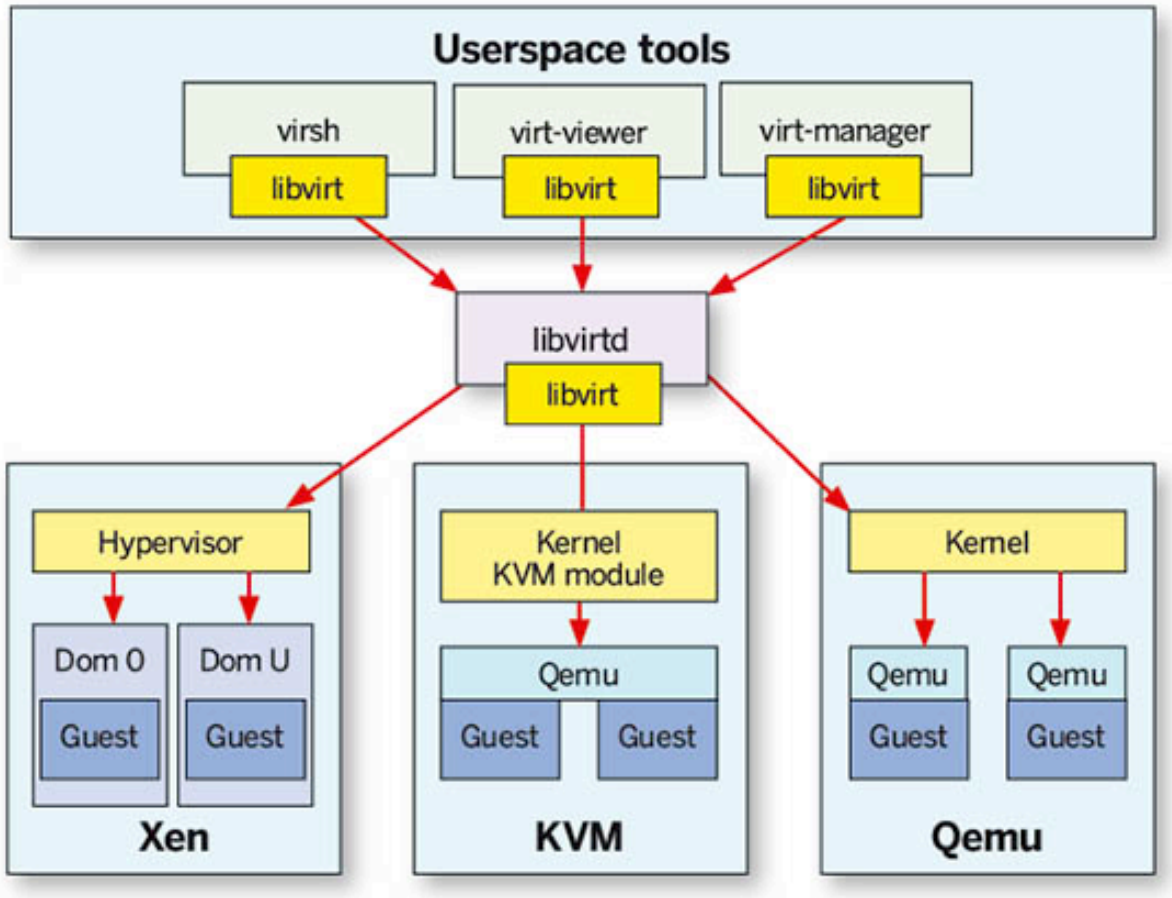
Libvirt-based tools
virt-viewer

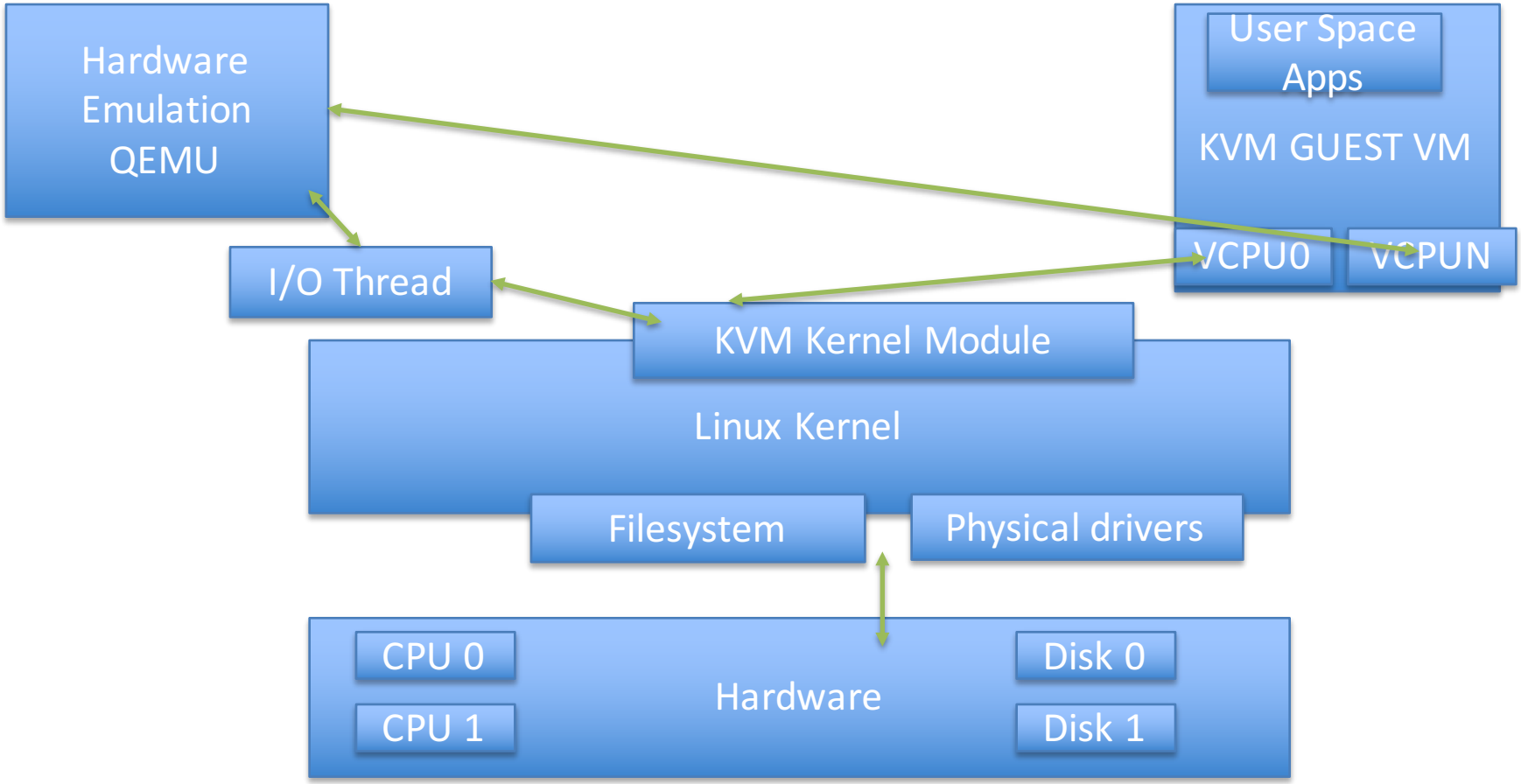
- Libvirt is a set of API libraries that sits in between the end user and the hypervisor
- The hypervisor could be built to use any virtualization technology such as KVM/QEMU, XEN, LXC, VirtualBox, VMWARE ESX, MS HyperV and even Parallels
- Libvirt acts as a sort of transparent layer that can take commands from users, modifies them based on the underlying virtualization technology, and then executes them on the actual hypervisor
- Tools include the libvirtd daemon, API library, and command line utility called virsh

Libvirt-based tools
virt-manager

- The Virtual Machine Manager is a desktop tool for managing VM Guests. It provides the ability to control the life cycle of existing machines (start/shutdown, pause/resume, save/restore) and create new VM Guests. It allows managing various types of storage and virtual networks. It provides access to the graphical console of VM Guests with a built-in VNC viewer and can be used to view performance statistics. virt-manager supports connecting to a local libvirtd, managing a local VM Host Server, or a remote libvirtd managing a remote VM Host Server.

Libvirt


Userspace tools
virsh
libvirt
virt-viewer
libvirt
virt-manager
libvirt
libvirtd
libvirt
Hypervisor
Dom 0
Guest
Dom U
Guest
Xen
Kernel
KVM module
Qemu
Guest
Guest
KVM
Kernel
Qemu
Guest
Qemu
Guest
Qemu


Hardware
Emulation
QEMU
User Space
Apps
KVM GUEST VM
VCPU0
VCPUN
I/O Thread
KVM Kernel Module
Linux Kernel
Filesystem
Physical drivers
CPU 0
CPU 1
Hardware
Disk 0
Disk 1

KVM Virtualization

KVM Setup

KVM Virtualization

System Requirements

KVM Virtualization

Installing Virtualization Packages

KVM Virtualization

Post Installation Task

KVM Virtualization

Managing Virtual Machines

KVM Virtualization

Using Virt-Manager

CentOS 7 Linux VM install

KVM Virtualization

Using Virt-Manager

Managing Instances

KVM Virtualization

Using virt-install

Installing a Windows Guest

KVM Virtualization

Disk Images

Using virt-builder

KVM Virtualization

Using virsh and virt-viewer

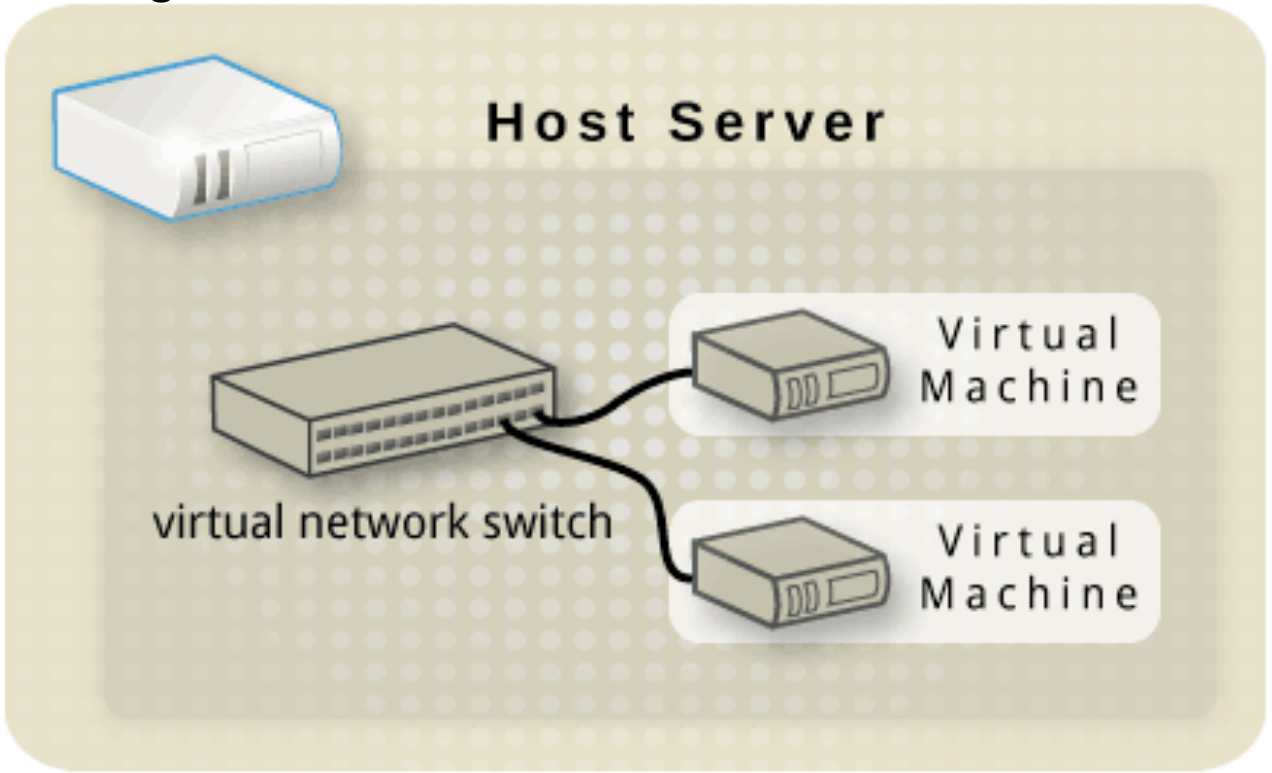
KVM Virtualization

KVM Network and Storage

KVM Virtualization

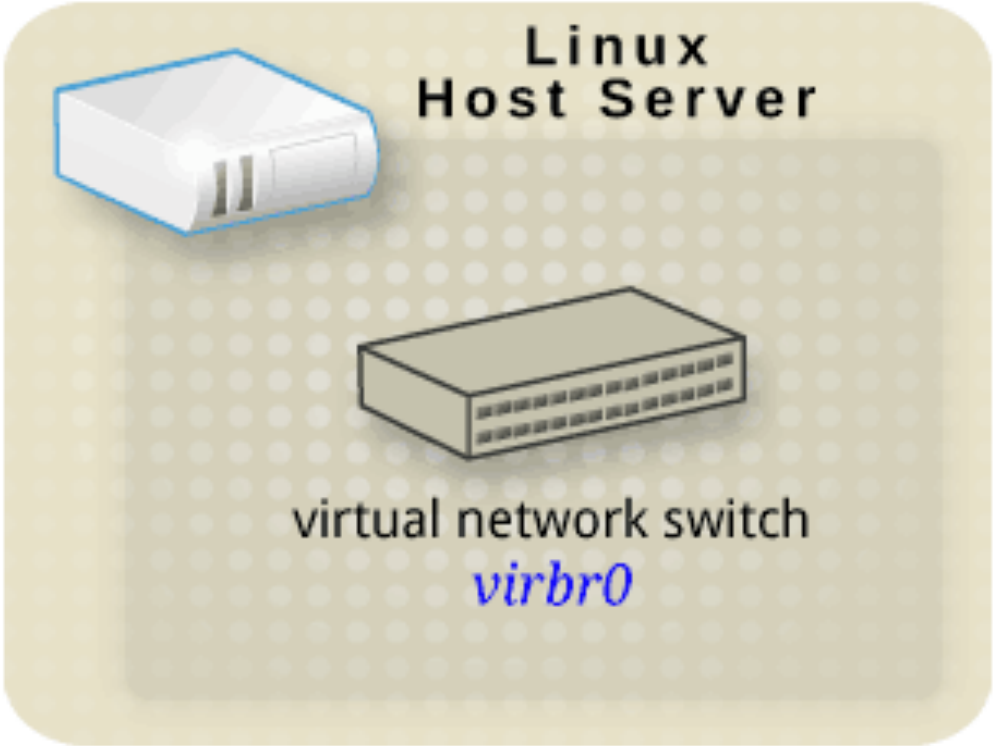
libvirt and virtual networking

Virtual Networking

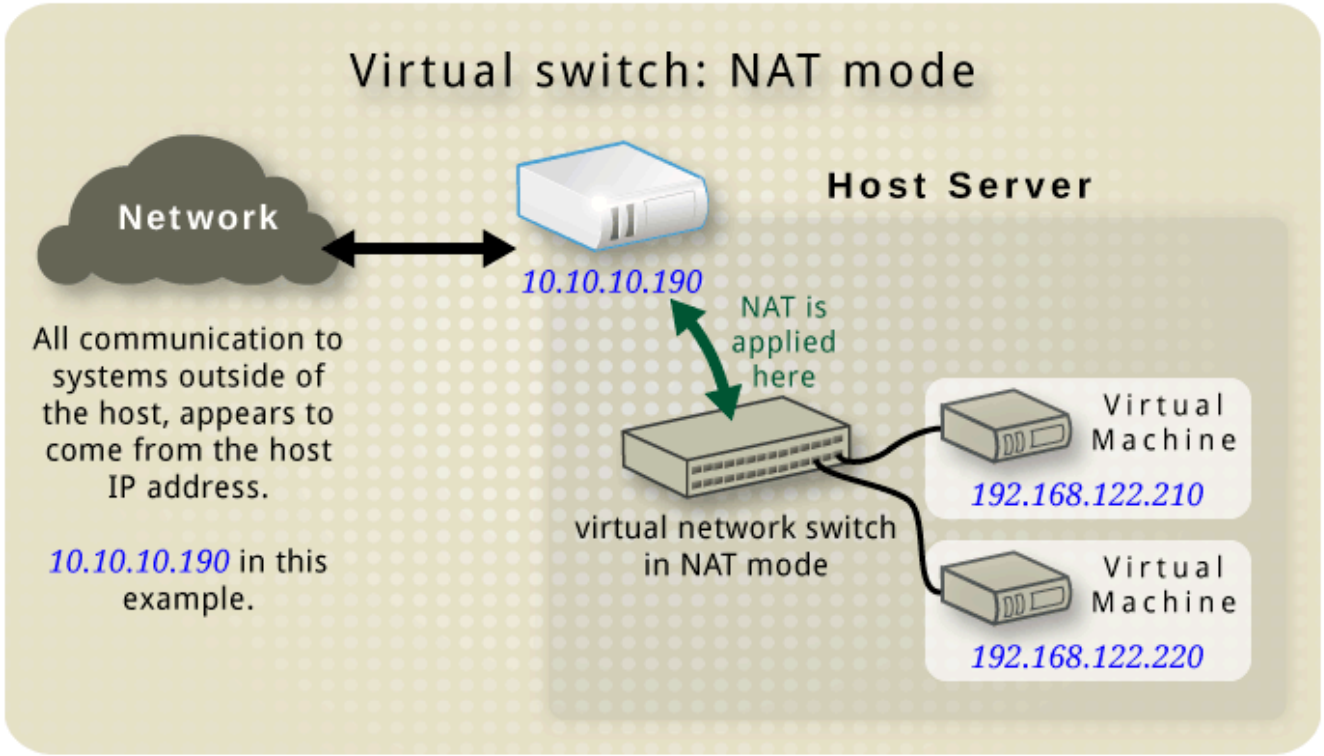


images from

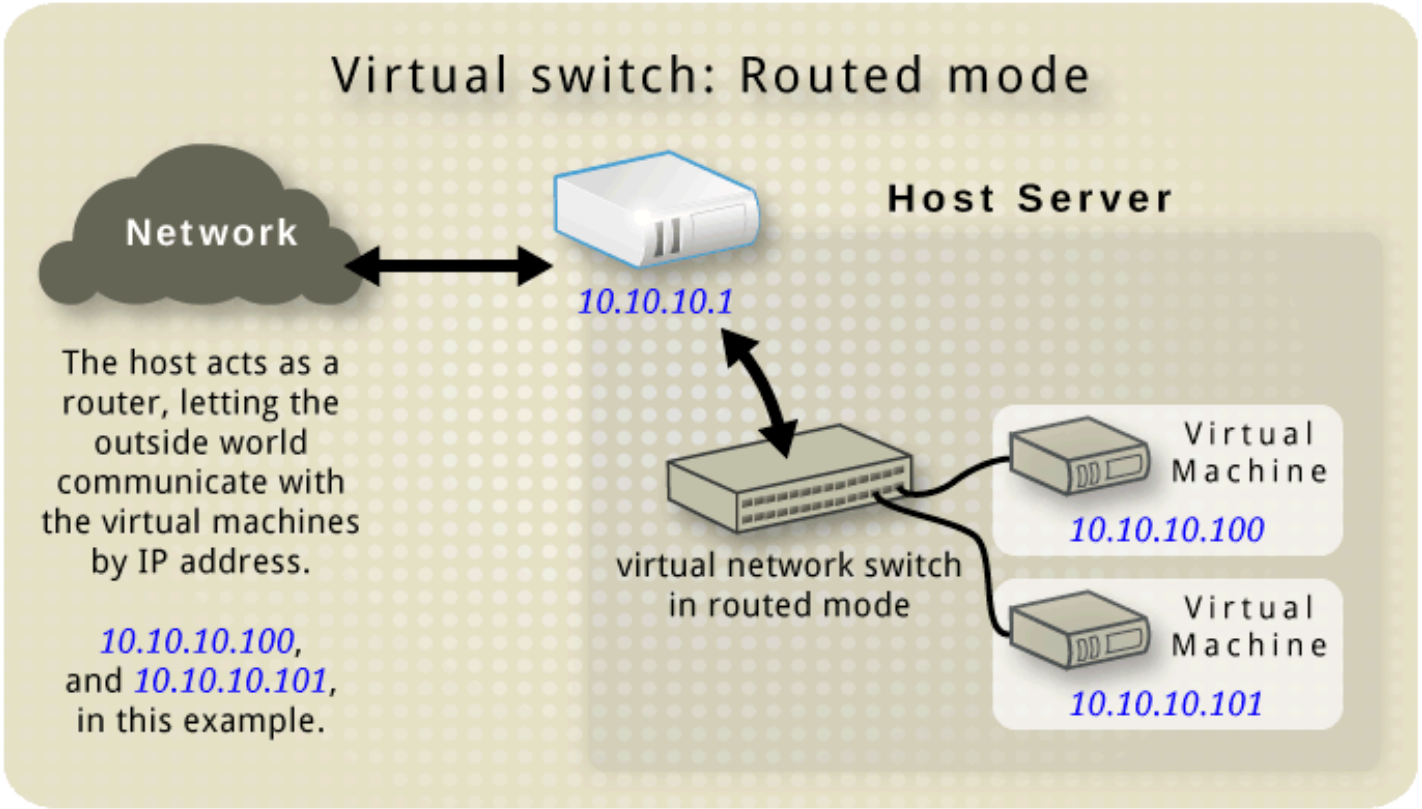
Virtual Networking



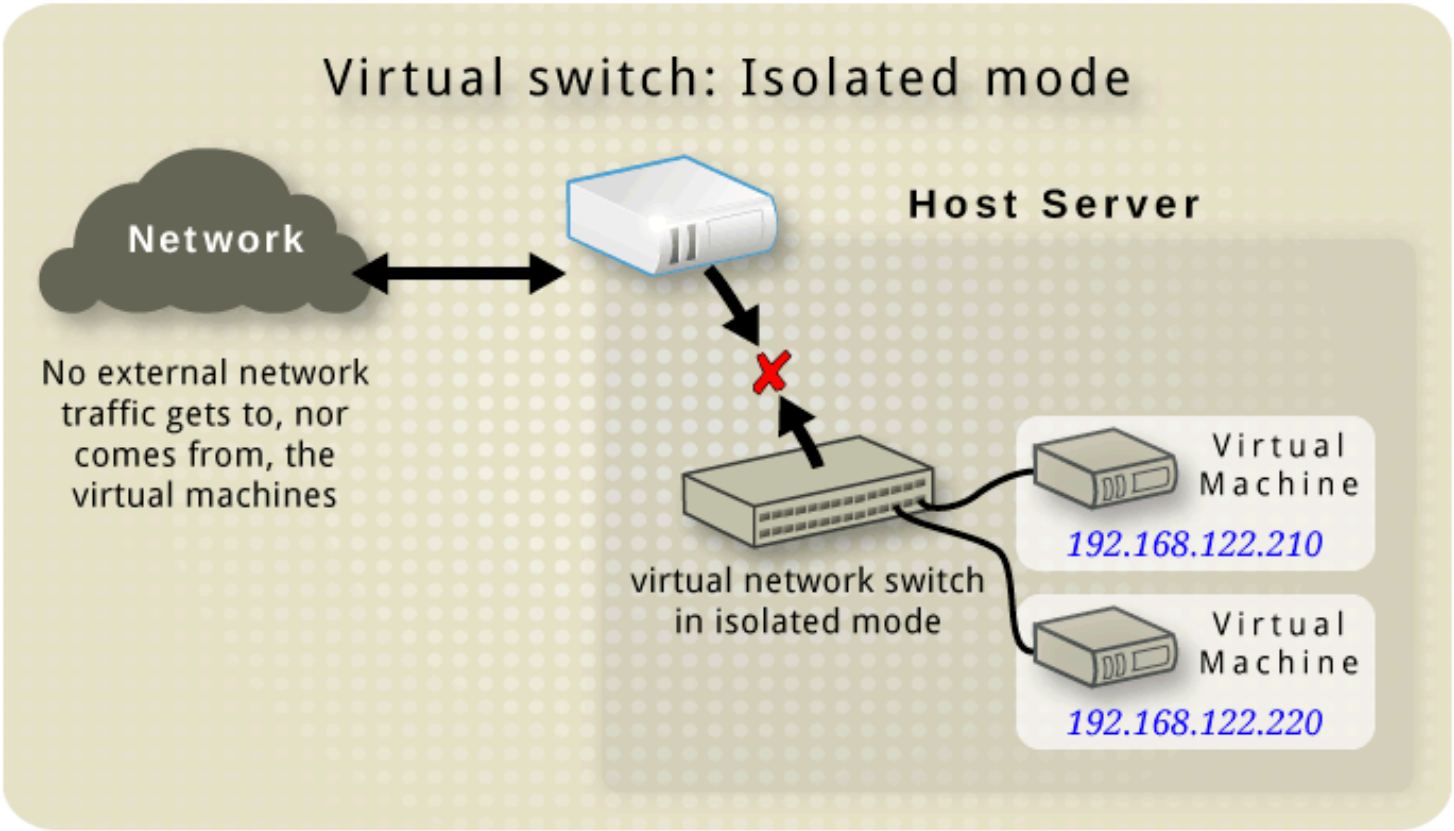
NAT Network Address Translation



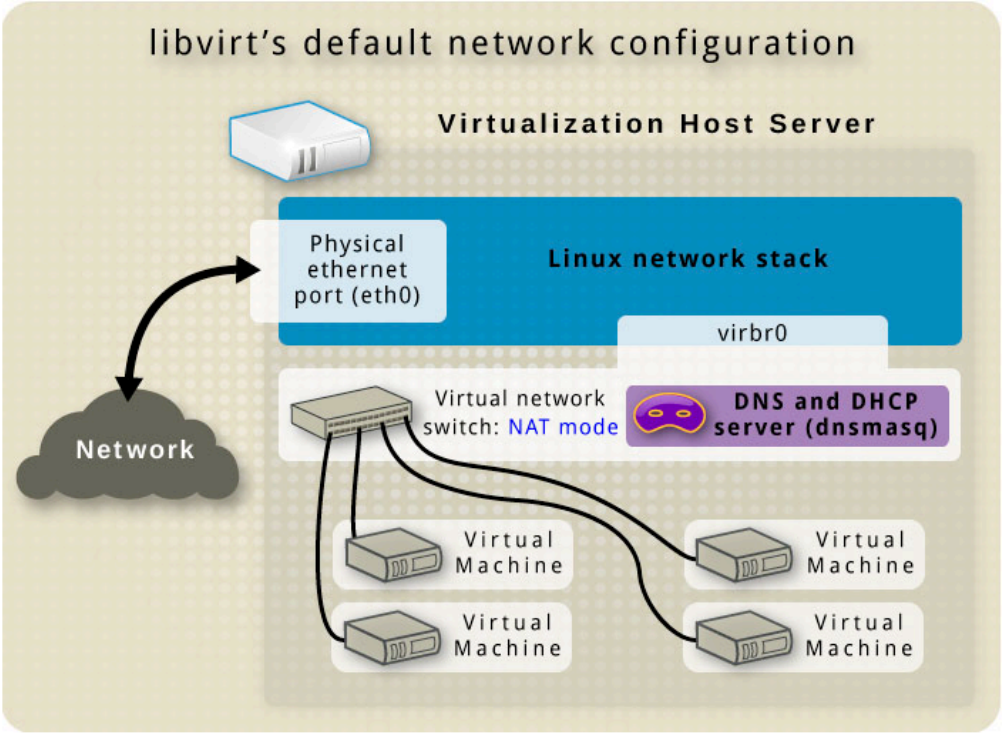
Routed Mode


Virtual switch: Routed mode
Network
Host Server
10.10.10.1
The host acts as a router, letting the outside world communicate with the virtual machines by IP address.
10.10.10.100, and 10.10.10.101, in this example.
virtual network switch in routed mode
Virtual Machine
10.10.10.100
Virtual Machine
10.10.10.101

Isolated Mode



Virtual Networking
libvirt's default networking configuration



Virtual Networking - MacVTap

KVM Virtualization

Creating an isolated network

KVM Virtualization

Creating a routed network

KVM Virtualization

Creating a Disk Image

Virtual Storage - Creating a disk image

KVM Virtualization

Storage Pools

KVM Virtualization

Backup and Recovery