

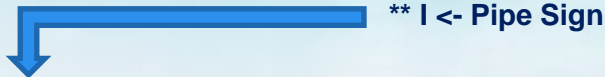


# KVM and Libvirt

Day 02, Session 2.3

- A hypervisor built into the Linux Kernel, based on QEMU
- It *requires* VT-x or AMD-V to run
- The host must be Linux

Checking



\*\* | <- Pipe Sign

```
egrep -c '(vmx|svm)' /proc/cpuinfo
```

If **0** it means that your CPU doesn't support hardware virtualization.

If **1** or more it does - but you still need to make sure that virtualization is enabled in the BIOS.

# Choosing a hypervisor

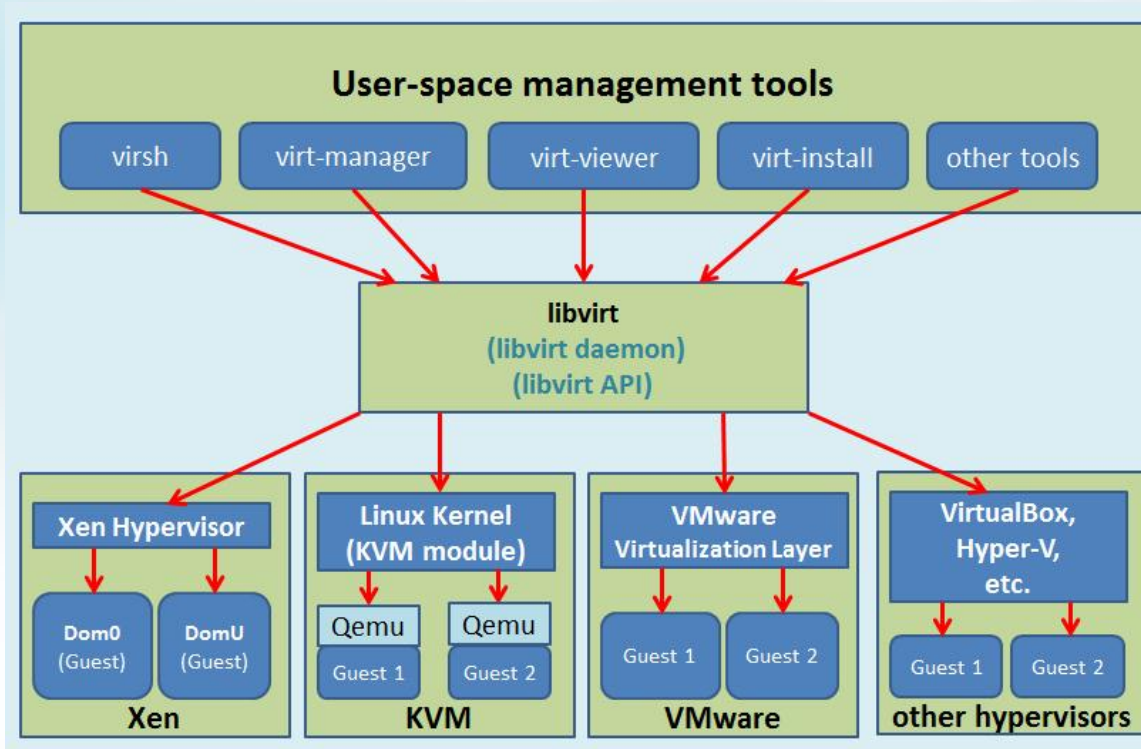
- **Step 1: Understand your needs**
  - Flexibility
  - Scalability
  - Availability
  - Reliability
  - Efficiency
  - Reliable support
- **Step 2: Understand the features**
- **Step 3: Investigate the ecosystem**
- **Step 4: Compare costs**
- **Step 5: Test for yourself**

- ❑ A toolkit to interact with virtualization software.
- ❑ A long term stable API.
- ❑ Supports a wide range of hypervisors

## Features:

- VM management
- Remote machine support
- Storage management
- Network interface management
- Virtual NAT and Route based networking

# Libvirt Process



- No simple web interface included
- virt-manager *can* talk to remote hypervisors, but virt-manager itself only runs under Linux
  - so you may end up running a VNC desktop into the Linux box, just to run virt-manager there
- XML format is unique to libvirt
  - different to OVF, VMX etc
  - too hard to write from scratch!
- libvirt's storage management is difficult

- **Virtual Machine Manager** application (**virt-manager** for short package name) is a desktop user interface for managing virtual machines. It presents a summary view of running domains, their live performance & resource utilization statistics. The detailed view graphs performance & utilization over time. Wizards enable the creation of new domains, and configuration & adjustment of a domain's resource allocation & virtual hardware. An embedded VNC client viewer presents a full graphical console to the guest domain.
- **virsh** is a command line interface tool for managing guests and the hypervisor.

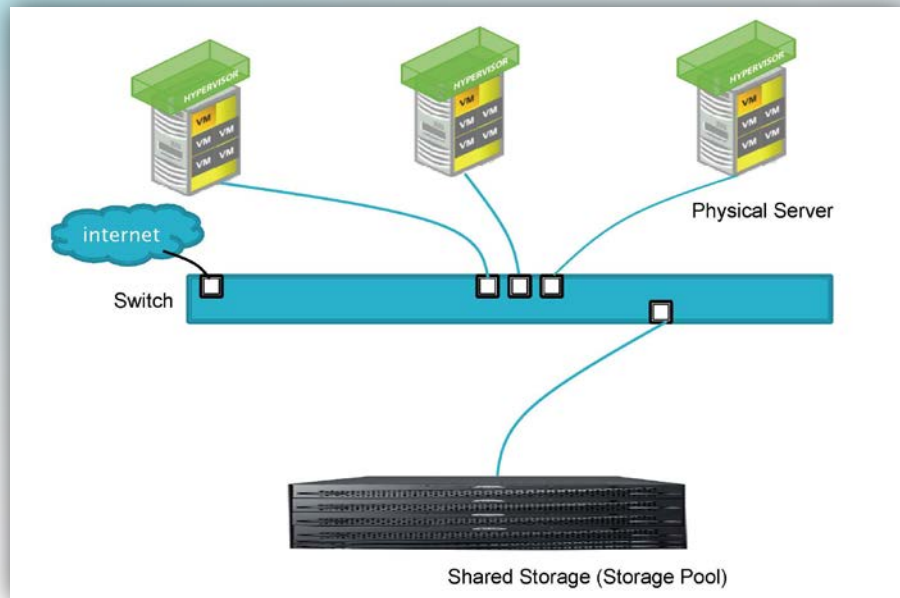
## install the packages

```
apt-get install bridge-utils qemu-kvm virtinst -y
```

- **Libvirt Main Clients**
  - virsh (Command Line)
  - virt-manager (Graphical X11)



## Concept



## Requirements

- The VM image is accessible on both source and destination hosts (located on a shared storage, e.g. using nfs).
- The src and dst hosts must be on the same subnet
- the guest on the destination must be started the same way it was started on the source.



**Any Questions ?**

**Thank you**