

主要分为三部分来学习：

0，简要介绍

1，环境部署

2，系统部署

3，批量克隆

第一部分：简要介绍

关于 **kvm**、**qemu**、**virt-tools**、**libvirt**

什么是 **KVM**？

KVM 是指基于 Linux 内核的虚拟机 (Kernel-based Virtual Machine)。2006 年 10 月，由以色列的 Qumranet 组织开发的一种新的“虚拟机”实现方案。2007 年 2 月发布的 Linux 2.6.20 内核第一次包含了 KVM。增加 KVM 到 Linux 内核是 Linux 发展的一个重要里程碑，这也是第一个整合到 Linux 主线内核的虚拟化技术。

KVM 在标准的 Linux 内核中增加了虚拟技术，从而我们可以通过优化的内核来使用虚拟技术。在 KVM 模型中，每一个虚拟机都是一个由 Linux 调度程序管理的标准进程，你可以在用户空间启动客户机操作系统。一个普通的 Linux 进程有两种运行模式：内核和用户。KVM 增加了第三种模式：客户模式（有自己的内核和用户模式）。

什么是 **libvirt**？

Libvirt 是库，是一种实现 Linux 虚拟化功能的 Linux? API，它支持各种虚拟机监控程序，包括 Xen 和 KVM，以及 QEMU 和用于其他操作系统的一些虚拟产品。

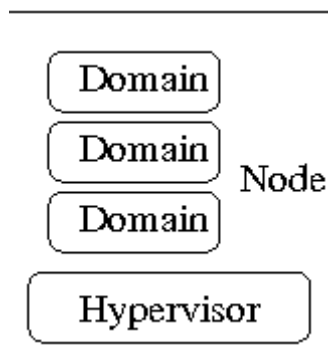
他提供一个命令和一个稳定的层来管理 node 上的域，甚至乎远程管理。

libvirt 提供所有 API 来做管理，例如提供，创建，修改，监控，控制，迁移，和停止域。

他的构成是这样的：

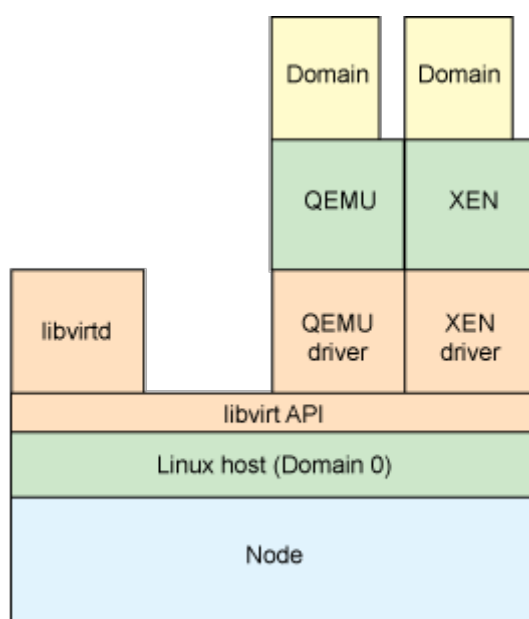
- a node is a single physical machine
- an hypervisor is a layer of software allowing to virtualize a node in a set of virtual machines with possibly different configurations than the node itself

- a domain is an instance of an operating system (or subsystem in the case of container virtualization) running on a virtualized machine provided by the hypervisor

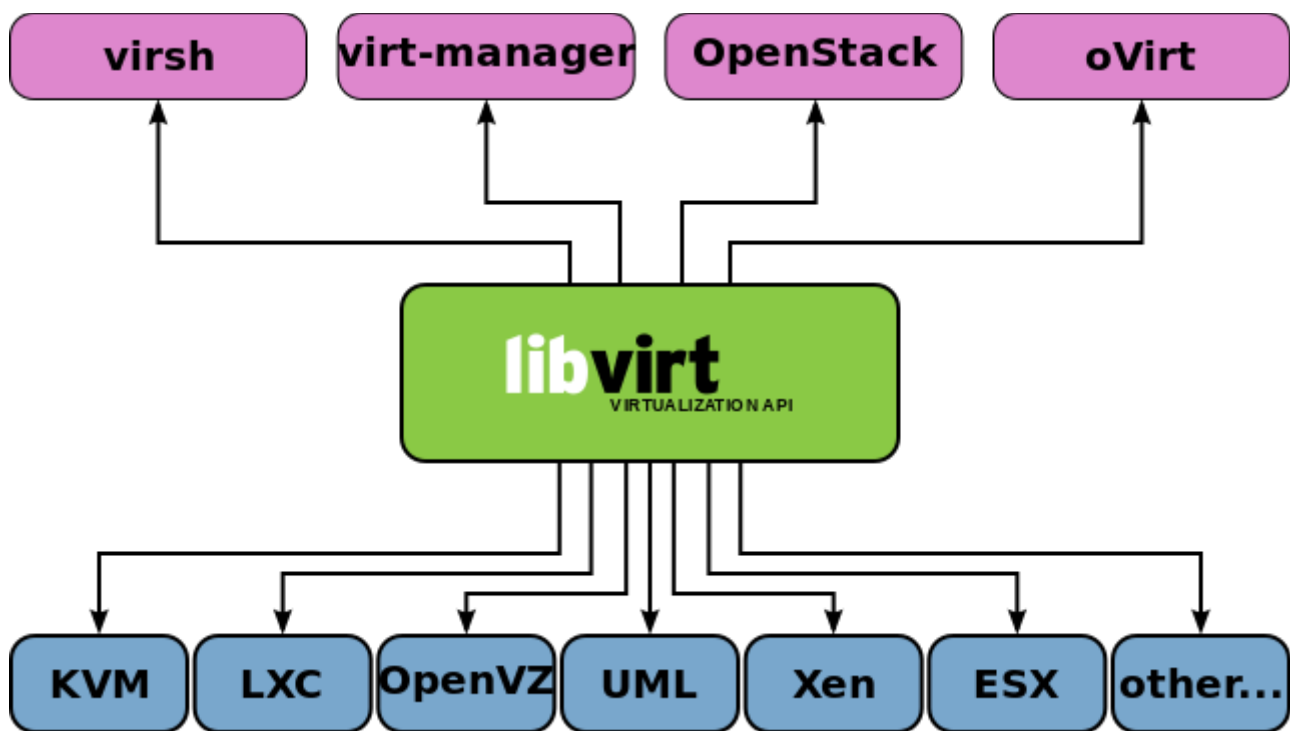


libvirt 把物理机叫节点 node，将虚拟客户机比作域 domain,还有一个 hypervisor 的软件负责虚拟化的（负责通讯和处理 api 请求等）

其实他的架构是这样的：



从另外一个层面来看是这样的：



引用参考：

- 1.<http://www.ibm.com/developerworks/cn/linux/l-libvirt/>
- 2.<http://en.wikipedia.org/wiki/Libvirt>
- 3.<http://libvirt.org/goals.html>

什么是 **qemu**

QEMU 是一套由 Fabrice Bellard 所编写的模拟处理器的自由软件。它与 Bochs，PearPC 近似，但其具有某些后两者所不具备的特性，如高速度及跨平台的特性。经由 kqemu 这个开源的加速器，QEMU 能模拟至接近真实电脑的速度。

进一步理解的话，kvm 只能虚拟 x86 平台的虚拟机，而 qemu 就可以虚拟 arm，mips 平台的虚拟机，这就是他们最大的区别。

引用参考：

- 1.http://wiki.qemu.org/Main_Page
- 2.<http://zh.wikipedia.org/wiki/QEMU>

什么是 **virt-tools**

virt-tools 是管理 libvirt 的工具，例如 virt-install virsh 等工具

引用参考：

1.<http://libvirt.org/virshcmdref.html>

表格一是列出各种不同的虚拟机程序的区别：

虚拟机监控程序	描述
Xen	面向 IA-32 , IA-64 和 PowerPC 970 架构的虚拟机监控程序
QEMU	面向各种架构的平台仿真器
Kernel-based Virtual Machine (KVM)	Linux 平台仿真器
Linux Containers (LXC)	用于操作系统虚拟化的 Linux (轻量级) 容器
OpenVZ	基于 Linux 内核的操作系统级虚拟化
VirtualBox	x86 虚拟化虚拟机监控程序
User Mode Linux	面向各种架构的 Linux 平台仿真器
Test	面向伪虚拟机监控程序的测试驱动器
Storage	存储池驱动器 (本地磁盘 , 网络磁盘 , iSCSI 卷)

第二部分：环境部署

环境检查

➤ 检查 CPU 信息

KVM 需要有 CPU 的支持 (Intel VT 或 AMD SVM) , 在安装 KVM 之前检查一下 CPU 是否提供了虚拟技术的支持。

基于 Intel 处理器的系统 , 运行 `grep vmx /proc/cpuinfo` 查找 CPU flags 是否包括 vmx 关键词。

基于 AMD 处理器的系统 , 运行 `grep svm /proc/cpuinfo` 查找 CPU flags 是否包括 svm 关键词

注：一些厂商禁止了机器 BIOS 中的 VT 选项，这种方式下 VT 不能被重新打开。

注意：/proc/cpuinfo 仅从 Linux 2.6.15(Intel) 和 Linux 2.6.16(AMD) 开始显示虚拟

化 方面的信息。请使用 `uname -r` 命令查询您的内核版本。如有疑问，请联系硬件厂商。

- 检查 BIOS，确保 BIOS 里开启 VT 选项

Intel(R) Virtualization Tech [Enabled]

如有必要，还需在 BIOS 中开启 VT 并重启机器。

- 关掉 networkmanager，因为这个会导致虚拟网卡配置冲突

```
service NetworkManager stop
```

```
chkconfig --del NetworkManager
```

- 关掉 selinux

修改/etc/selinux/config 文件

将 SELINUX=enforcing 改为 SELINUX=disabled

```
setenforce 0
```

环境安装

- `yum -y install kvm python virtinst libvirt tuncctl bridge utils virt manager qemu kvm tools virt-viewer virt v2v`

or

- `yum install kvm virt * libvirt`

安装完成后最好重启服务器

- 启动 libvirtd 服务

```
service libvirtd start
```

```
chkconfig --level 3 libvirtd on
```

```
virsh list
```

- 检查是否可以使用 kvm 模块

```
lsmod | grep kvm
```

```
kvm_intel          54285  9
```

```
kvm                333172  1 kvm_intel
```

网络配置

➤ NAT forwarding

```
# virsh net-list --all
Name                               State      Autostart
-----
default                           active     yes
```

如果结果不是上面的样子，做如下操作

```
# virsh net-define /usr/share/libvirt/networks/default.xml
Network default defined from /usr/share/libvirt/networks/default.xml
# virsh net-autostart default
Network default marked as autostarted
# virsh net-start default
Network default started
```

```
# brctl show
bridge name      bridge id        STP enabled    interfaces
virbr0           8000.000000000000 yes
```

记得开启转发

```
net.ipv4.ip_forward = 1
```

➤ 桥接模式

进入网卡配置目录

```
cd /etc/sysconfig/network-scripts/
```

```
cp ifcfg-eth0 ifcfg-br0
```

```
# cat ifcfg-eth0
DEVICE=eth0
HWADDR=BC:EE:7B:9A:5E:91
TYPE=Ethernet
UUID=a257d75d-072c-4eac-8daa-b131324fff52
ONBOOT=yes
NM_CONTROLLED=yes
```

DNS1=8.8.8.8

BRIDGE=br0

cat ifcfg-br0

DEVICE=br0

HWADDR=BC:EE:7B:9A:5E:91

UUID=a257d75d-072c-4eac-8daa-b131324fff52

ONBOOT=yes

TYPE=Bridge

NM_CONTROLLED=yes

BOOTPROTO=static

IPADDR=192.168.0.12

NETMASK=255.255.255.0

GATEWAY=192.168.0.1

DNS1=8.8.8.8

配置完成后 执行

service network restart 即可看到效果

说明：

- 1.DNS 服务器配置随意
- 2.br0 是虚拟网卡，libvirt 是安装这个软件后出来的
- 3.实体网卡去掉相关 ip 属性，如 BOOTPROTO IPADDR NETMASK GATEWAY,虚拟网卡将这些属性补充到自己的网卡配置文件里面去，但其他配置一律不变动，相当于 br0 变成实体网卡了，但只是相当于而已。
- 4.虚拟网卡需要将 TYPE 改为 Bridge，证明是使用桥接模式。

安装虚拟机--创建磁盘

创建磁盘文件不能太小，不然会 coredump。

qcow2 格式是 kvm 支持的标准格式，raw 格式为虚拟磁盘文件通用格式。有测试数据表明 raw 格式的 I/O 性能略高于 qcow2 格式，但是在加密，容量，快照方面 qcow2 格式有优势

```
qemu-img create -f qcow2 test.qcow2 20G //建立 qcow2 格式磁盘文件
qemu-img create -f raw test.raw 20G //建立 raw 格式磁盘文件
qemu-img info test.qcow2 //查看已经创建的虚拟磁盘文件
```

安装虚拟机—系统部署

➤ 以下是 ubuntu 为例，ubuntu 14

```
cat vir_ubuntu.sh
#!/bin/bash
virt-install --name ubuntu \
--boot network,cdrom,menu=on \
--ram 512 --vcpus=1 \
--os-variant=generic \
--cdrom=/app/ftp_data/team/public/system_iso/ubuntu-14.04-desktop-
i386.iso \
--disk
path=/app/kvm/images/centos.qcow2,size=10,format=qcow2,bus=virtio
\
--bridge=br0,model=virtio --keymap=en-us \
--vnc --vncport=5910 --vnclisten=0.0.0.0
```

➤ 以下是 centos 6.5 为例

```
cat vir_centos.sh
#!/bin/bash
virt-install --name centos6.5x64 \
--boot network,cdrom,menu=on \
--ram 512 --vcpus=1 \
--os-variant=generic \
--cdrom=/app/ftp_data/team/public/system_iso/CentOS-6.5-x86_64-
bin-DVD1.iso \
```



```
--disk path=/app/ftp_data/team/public/system_iso/another_CentOS-6.5-x86_64-bin-DVD1.iso,device=cdrom \  
--disk  
path=/app/kvm/images/centos.qcow2,size=10,format=qcow2,bus=virtio \  
--bridge=br0,model=virtio --keymap=en-us \  
--vnc --vncport=5911 --vnclisten=0.0.0.0
```

➤ 以下是 xp 系统为例：

```
cat vir_xp2.sh  
virt-install \  
--name=xp2 \  
--ram 256 \  
--vcpus=1 \  
--disk path=/app/kvm/images/xp2.img,size=10 \  
--cdrom=/app/ftp_data/team/public/system_iso/zh-hans_windows_xp_professional_with_service_pack_3_x86_cd_v1_x14-74070.iso \  
--vnc --vncport=5913 --vnclisten=0.0.0.0 \  
--network bridge=br0,model=virtio \  
--noautoconsole \  
--os-variant=winxp \  
--keymap=en-us
```

说明：

1.os-variant 是指支持什么虚拟机，以下是支持的类型，按需选择即可

win7	: Microsoft Windows 7
vista	: Microsoft Windows Vista
winxp64	: Microsoft Windows XP (x86_64)
winxp	: Microsoft Windows XP
win2k8	: Microsoft Windows Server 2008
win2k3	: Microsoft Windows Server 2003

freebsd8	: FreeBSD 8.x
generic	: Generic
debiansqueeze	: Debian Squeeze
debianlenny	: Debian Lenny
fedora16	: Fedora 16
fedora15	: Fedora 15
fedora14	: Fedora 14
mes5.1	: Mandriva Enterprise Server 5.1 and later
mandriva2010	: Mandriva Linux 2010 and later
rhel6	: Red Hat Enterprise Linux 6
rhel5.4	: Red Hat Enterprise Linux 5.4 or later
rhel4	: Red Hat Enterprise Linux 4
sles11	: Suse Linux Enterprise Server 11
sles10	: Suse Linux Enterprise Server
ubuntuneiric	: Ubuntu 11.10 (Oneiric Ocelot)
ubuntunatty	: Ubuntu 11.04 (Natty Narwhal)
ubuntumaverick	: Ubuntu 10.10 (Maverick Meerkat)
ubuntulucid	: Ubuntu 10.04 (Lucid Lynx)
ubuntuhardy	: Ubuntu 8.04 LTS (Hardy Heron)

2.disk 属性就是指虚拟机使用哪一个磁盘镜像文件作为存储地方，指定刚才创建的镜像文件位置即可，size=10,format=qcow2,bus=virtio 分别是指大小，格式（跟镜像格式一致），磁盘总线类型（指 Value can be 'ide', 'scsi', 'usb', 'virtio' or 'xen', virtio 是专门为 kvm 制定的 bus，比较常用），大小需要比镜像文件大小要小，相当于镜像是容器，这个 disk 是里面的水，水太多就溢出了。

3.bridge=br0,model=virtio -keymap=en-us 分别指网卡使用，网卡模式即驱动模式，和键盘制式，桥接模式都使用 br0，而虚拟机需要使用 virtio 这个驱动才能使用虚拟网卡，这个虚拟网卡驱动在 linux 下支持很好，在 windows 下支持一般，键盘制式如果不指定的话会出现键位错乱的情况，一般我们使用英式。

4.安装 centos6.5 的时候我使用了 2 次 disk，原因是因为 centos6.5 的安装盘启动引导安装有问题，会出现无法查找到安装源的情况，所以特地加载多一个 iso，一个作为启动使用，一个作为安装源使用，作为安装源使用的时候是使用磁盘格式存储。

5.vnc 相关配置，这个 vnc 如果配置了的话是 virt-tools 组件自带的，无需要额外安装 vnc 服务端，默认绑定 127.0.0.1，这里将其改为 0.0.0.0 以便可以通过外部连接，每一台的虚拟机的端口都不能一致，一致就会冲突而造成无法启动安装。

6.如果无须使用图形界面安装，virt-install 默认是支持文本模式安装的。

实例：安装 xp

➤ 创建磁盘

```
[KALI root@kali /home/virst]
#cat create_disk.sh
#!/bin/bash
#####
# ScriptName: create_disk.sh
# Author: liujmsunits@hotmail.com
# Create Date: 2016-04-02 09:49
# Modify Author: liujmsunits@hotmail.com
# Modify Date: 2016-04-13 08:02
# Function:
#####
qemu-img create -f qcow2 ../vhost/xp2.qcow2 10G  #//建立qcow2格式磁盘文件
#qemu-img create -f raw test.raw 8G             #//建立raw格式磁盘文件
qemu-img info ../vhost/xp2.qcow2                #//查看已经创建的虚拟磁盘文件
```

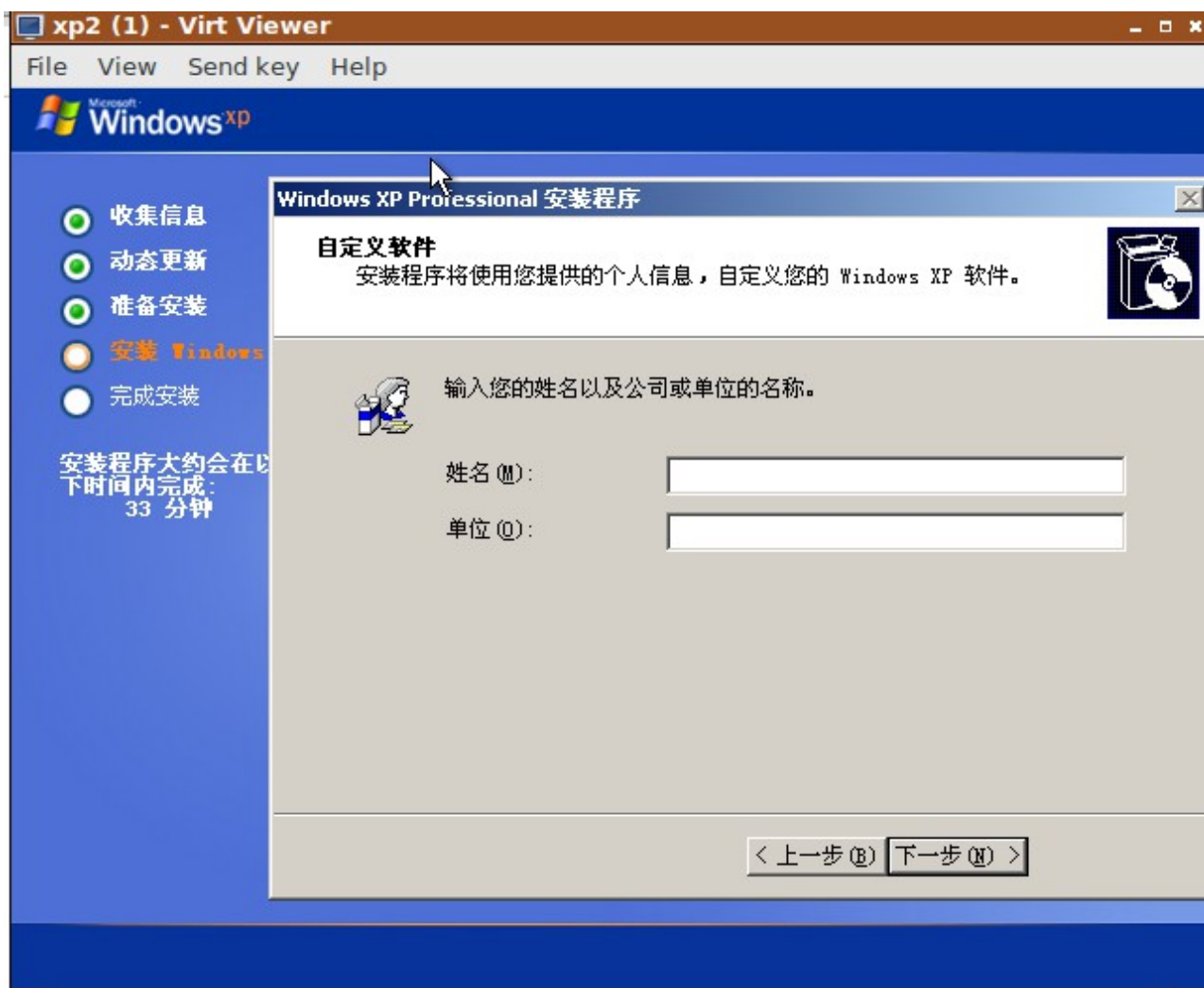
➤ 安装系统

```
[KALI root@kali /home/virst]
#cat virt_xp.sh
#!/bin/bash
#####
# ScriptName: create_virsh_xp.sh
# Author: liujmsunits@hotmail.com
# Create Date: 2016-04-12 22:25
# Modify Author: liujmsunits@hotmail.com
# Modify Date: 2016-04-13 08:02
# Function:
#####
virt-install \
    --name=xp2 \
    --ram 512 \
    --vcpus=1 \
    --disk path=/home/vhost/xp2.qcow2,size=10 \
    --cdrom=/virtualization/zh-hans_windows_xp_professional_with_service_pack_3_x86_cd_vl_x14-74070.iso \
    --vnc --vncport=5913 --vnclisten=0.0.0.0 \
    --network network='default',model=virtio \
    --noautoconsole \
    --os-variant=winxp \
    --keymap=en-us
```

```
root@kali:/home/virst# sh virt_xp.sh
Starting install...
Creating domain...
Domain installation still in progress. Waiting for installation to complete.
Domain has shutdown. Continuing.
Starting domain...
Domain installation still in progress. Waiting for installation to complete.
```

➤ 连接到图像界面

```
[KALI root@kali /home/virst]
#cat connect.sh
#!/bin/bash
#*****#
# ScriptName: connect.sh
# Author: liujmsunits@hotmail.com
# Create Date: 2016-04-01 22:54
# Modify Author: liujmsunits@hotmail.com
# Modify Date: 2016-04-13 08:05
# Function:
#*****#
/usr/bin/virt-viewer --connect qemu:///system --wait xp2
```



MRX3F-47B9T-2487J-KWKMF-RPWBY

镜像下载网址：www.itellyou.cn

参考网站：

[http://www.techotopia.com/index.php/Installing_a_CentOS_KVM_Guest_OS_from_the_Command-line_\(virt-install\)](http://www.techotopia.com/index.php/Installing_a_CentOS_KVM_Guest_OS_from_the_Command-line_(virt-install))

http://www.godblessyuan.com/2015/01/20/kvm_use_thinking/