

本站文章大部分为作者原创，非商业用途转载无需作者授权，但务必在文章标题下面注明作者 刘世民（Sammy Liu）以及可点击的本博客地址超级链接 <http://www.cnblogs.com/sammyliu/>，谢谢合作



世民谈云计算

（声明：本站文章皆基于公开来源信息，仅代表作者个人观点，与作者所在公司无关）

昵称：SammyLiu
园龄：2年6个月
荣誉：推荐博客
粉丝：470
关注：30
[+加关注](#)

< 2017年5月 >						
日	一	二	三	四	五	六
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

常用链接

- [我的随笔](#)
- [我的评论](#)
- [我的参与](#)
- [最新评论](#)
- [我的标签](#)

我的标签

- [GRE\(1\)](#)
- [Neutron\(1\)](#)
- [Open vSwitch\(1\)](#)
- [OpenStack\(1\)](#)

随笔分类(254)

- [Ceilometer\(3\)](#)
- [Ceph\(13\)](#)
- [Cinder\(6\)](#)
- [Docker\(8\)](#)
- [Glance](#)
- [Heat\(2\)](#)
- [K8S](#)
- [Keystone\(1\)](#)
- [KVM\(10\)](#)
- [MessageQueue\(4\)](#)
- [MySQL\(1\)](#)
- [Neutron\(17\)](#)
- [Nova\(10\)](#)
- [OpenStack\(33\)](#)
- [Sahara](#)
- [Storage\(1\)](#)
- [Swift\(3\)](#)
- [Trove](#)
- [Ubuntu\(3\)](#)
- [VMware\(3\)](#)
- [安装和配置\(1\)](#)
- [版本\(4\)](#)
- [备份\(1\)](#)
- [大数据\(5\)](#)
- [翻译\(4\)](#)
- [高可用（HA）\(6\)](#)
- [基础知识\(19\)](#)
- [监控\(1\)](#)
- [容器\(4\)](#)
- [容器编排](#)
- [使用案例\(4\)](#)
- [网络\(8\)](#)

[博客园](#) [首页](#) [新随笔](#) [订阅](#) [XML](#) [管理](#)

随笔-121 评论-504 文章-45

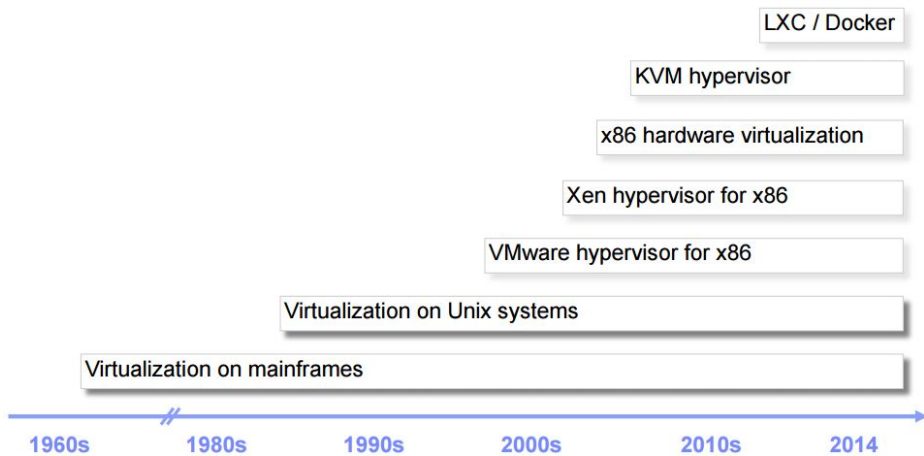
KVM 介绍（1）：简介及安装

学习 KVM 的系列文章：

- [（1）介绍和安装](#)
- [（2）CPU 和 内存虚拟化](#)
- [（3）I/O QEMU 全虚拟化和准虚拟化（Para-virtualization）](#)
- [（4）I/O PCI/PCIe设备直接分配和 SR-IOV](#)
- [（5）libvirt 介绍](#)
- [（6）Nova 通过 libvirt 管理 QEMU/KVM 虚拟机](#)
- [（7）快照（snapshot）](#)
- [（8）迁移（migration）](#)

1. KVM 介绍

1.0 虚拟化简史



其中，KVM 全称是 基于内核的虚拟机（Kernel-based Virtual Machine），它是一个 Linux 的一个内核模块，该内核模块使得 Linux 变成了一个 Hypervisor：

- 它由 Qumrnet 开发，该公司于 2008 年被 Red Hat 收购。
- 它支持 x86 (32 and 64 位), s390, Powerpc 等 CPU。
- 它从 Linux 2.6.20 起就作为一模块被包含在 Linux 内核中。
- 它需要支持虚拟化扩展的 CPU。
- 它是完全开源的。[官网](#)。

本文介绍的是基于 X86 CPU 的 KVM。

1.1 KVM 架构

KVM 是基于虚拟化扩展（Intel VT 或者 AMD-V）的 X86 硬件的开源的 Linux 原生的全虚拟化解决方案。KVM 中，虚拟机被实现为常规的 Linux 进程，由标准 Linux 调度程序进行调度；虚拟机的每个虚拟 CPU 被实现为一个常规的 Linux 进程。这使得 KVM 能够使用 Linux 内核的已有功能。

但是，KVM 本身不执行任何硬件模拟，需要客户空间程序通过 /dev/kvm 接口设置一个客户机虚拟服务器的地址空间，向它提供模拟的 I/O，并将它的视频显示映射回宿主的显示屏。目前这个应用程序是 QEMU。

Linux 上的用户空间、内核空间和虚拟机：

问题定位(3)

行业(14)

性能(4)

虚拟化(7)

原理(22)

云Cloud(29)

随笔档案(121)

2017年5月 (1)

2017年3月 (1)

2017年1月 (1)

2016年10月 (7)

2016年9月 (5)

2016年8月 (4)

2016年7月 (1)

2016年6月 (5)

2016年5月 (1)

2016年4月 (1)

2016年3月 (9)

2016年2月 (4)

2016年1月 (2)

2015年12月 (7)

2015年11月 (7)

2015年10月 (4)

2015年9月 (4)

2015年8月 (5)

2015年7月 (9)

2015年6月 (10)

2015年5月 (3)

2015年4月 (11)

2015年3月 (2)

2015年2月 (6)

2015年1月 (5)

2014年12月 (6)

文章分类(21)

Ceph(1)

GlusterFS

Web 服务器(2)

操作系统(1)

存储

大数据(2)

分布式系统

服务器(1)

网络(11)

虚拟化(3)

云

文章档案(42)

2016年10月 (2)

2016年9月 (1)

2016年6月 (1)

2016年5月 (3)

2015年12月 (4)

2015年10月 (5)

2015年9月 (2)

2015年6月 (1)

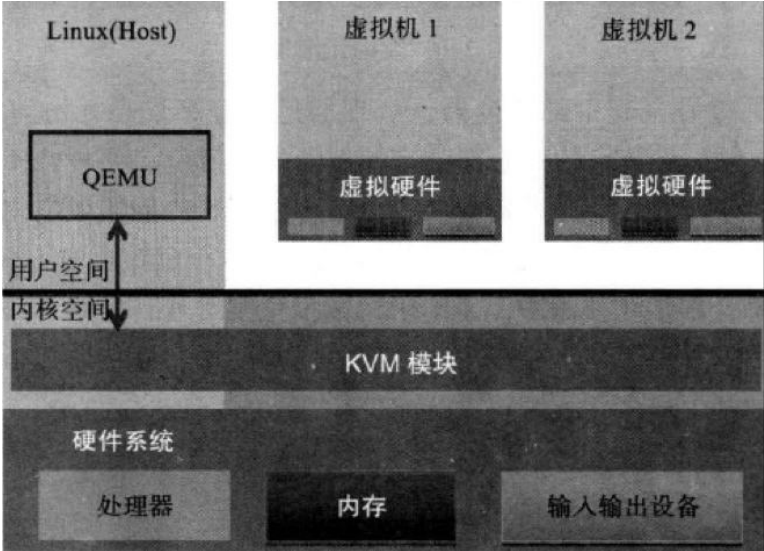
2015年4月 (23)

积分与排名

积分 - 286831

排名 - 535

最新评论



- Guest：客户机系统，包括CPU（vCPU）、内存、驱动（Console、网卡、I/O 设备驱动等），被 KVM 置于一种受限制的 CPU 模式下运行。
- KVM：运行在内核空间，提供CPU 和内存的虚拟化，以及客户机的 I/O 拦截。Guest 的 I/O 被 KVM 拦截后，交给 QEMU 处理。
- QEMU：修改过的为 KVM 虚拟机使用的 QEMU 代码，运行在用户空间，提供硬件 I/O 虚拟化，通过 IOCTL /dev/kvm 设备和 KVM 交互。

KVM 是实现拦截虚拟机的 I/O 请求的原理：

现代 CPU 本身了对特殊指令的截获和重定向的硬件支持，甚至新的硬件会提供额外的资源来帮助软件实现对关键硬件资源的虚拟化从而提高性能。以 X86 平台为例，支持虚拟化技术的 CPU 带有特别优化过的指令集来控制虚拟化过程。通过这些指令集，VMM 很容易将客户机置于一种受限制的模式下运行，一旦客户机试图访问物理资源，硬件会暂停客户机的运行，将控制权交还给 VMM 处理。VMM 还可以利用硬件的虚拟化增强机制，将客户机在受限模式下对一些特定资源的访问，完全由硬件重定向到 VMM 指定的虚拟资源，整个过程不需要暂停客户机的运行和 VMM 的参与。由于虚拟化硬件提供全新的架构，支持操作系统直接在上面运行，无需进行二进制转换，减少了相关的性能开销，极大简化了VMM的设计，使得VMM性能更加强大。从 2005 年开始，Intel 在其处理器产品线中推广 Intel Virtualization Technology 即 IntelVT 技术。

QEMU-KVM：

其实 QEMU 原本不是 KVM 的一部分，它自己就是一个纯软件实现的虚拟化系统，所以其性能低下。但是，QEMU 代码中包含整套的虚拟机实现，包括处理器虚拟化，内存虚拟化，以及 KVM需要使用到的虚拟设备模拟（网卡、显卡、存储控制器和硬盘等）。为了简化代码，KVM 在 QEMU 的基础上做了修改。VM 运行期间，QEMU 会通过 KVM 模块提供的系统调用进入内核，由 KVM 负责将虚拟机置于处理的特殊模式运行。遇到虚拟机进行 I/O 操作，KVM 会从上次的系统调用出口处返回 QEMU，由 QEMU 来负责解析和模拟这些设备。从 QEMU 的角度看，也可以说是 QEMU 使用了 KVM 模块的虚拟化功能，为自己的虚拟机提供了硬件虚拟化加速。除此以外，虚拟机的配置和创建、虚拟机运行所依赖的虚拟设备、虚拟机运行时的用户环境和交互，以及一些虚拟机的特定技术比如动态迁移，都是 QEMU 自己实现的。

KVM：

KVM 内核模块在运行时按需加载进入内核空间运行。KVM 本身不执行任何设备模拟，需要 QEMU 通过 /dev/kvm 接口设置一个 GUEST OS 的地址空间，向它提供模拟的 I/O 设备，并将它的视频显示映射回宿主机的显示屏。它是KVM 虚拟机的核心部分，其主要功能是初始化 CPU 硬件，打开虚拟化模式，然后将虚拟客户机运行在虚拟机模式下，并对虚拟机的运行提供一定的支持。以在 Intel 上运行为例，KVM 模块被加载的时候，它：

1. 首先初始化内部的数据结构；
2. 做好准备后，KVM 模块检测当前的 CPU，然后打开 CPU 控制及存取 CR4 的虚拟化模式开关，并通过执行 VMXON 指令将宿主操作系统置于虚拟化模式的根模式；
3. 最后，KVM 模块创建特殊设备文件 /dev/kvm 并等待来自用户空间的指令。

接下来的虚拟机的创建和运行将是 QEMU 和 KVM 相互配合的过程。两者的通信接口主要是一系列针对特殊设备文件 dev/kvm 的 IOCTL 调用。其中最重要的是创建虚拟机。它可以理解成KVM 为了某个特定的虚拟机创建对应的内核数据结构，同时，KVM 返回一个文件句柄来代表所创建的虚拟机。

针对该句柄的调用可以对虚拟机做相应地管理，比如创建用户空间虚拟地址和客户机物理地址、真实物理地址之间的映射关系，再比如创建多个 vCPU。KVM 为每一个 vCPU 生成对应的文件句柄，对其相应地 IOCTL 调用，就可以对vCPU进行管理。其中最重要的就是“执行虚拟处理器”。通过它，虚拟机在 KVM 的支持下，被置于虚拟化模式的非根模式下，开始执行二进制指令。在非根模式下，所有敏感的二进制指令都被CPU捕捉到，CPU 在保存现场之后自动切换到根模式，由 KVM 决定如何处理。

除了 CPU 的虚拟化，内存虚拟化也由 KVM 实现。实际上，内存虚拟化往往是一个虚拟机实现中最复杂的部分。CPU 中的内存管理单元 MMU 是通过页表的形式将程序运行的虚拟地址转换成实际物理地址。在虚拟机模式下，MMU 的页表则必须在一次查询的时候完成两次地址转换。因为除了将客户机程序的虚拟地址转换了客户机的物理地址外，还要将客户机物理地址转化成真实物理地址。

2. KVM 的功能列表

KVM 所支持的功能包括：

- 支持CPU 和 memory 超分（Overcommit）

1. Re:Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network]

eth1 - 公共网络 (untagged), 管理网络 (tag=102), 存储网络 (tag=103) 不好意思, 大家共用同一个eth1端口的时候, 请问这里交换机端口是配置为tagged还是untagged.....

--xianke9

2. Re:理解Docker (5) : Docker 网络

1.12版本上网络的表现如何?

--幽灵狼

3. Re:理解Docker (5) : Docker 网络

我想请问一下运行docker quickstart terminal时一直卡在“waiting for an IP”应该如何解决呢? 希望楼主能解答一下。

--silentbell

4. Re:理解Docker (6) : 若干企业生产环境中的容器网络方案写得好好! 加油。

--itbj00

5. Re:理解Docker (5) : Docker 网络

非常好, 写得很详细。加油!

--itbj00

阅读排行榜

1. Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network](22087)
2. 理解 OpenStack 高可用 (HA) (1) : OpenStack 高可用和灾备方案 [OpenStack HA and DR](13707)
3. Neutron 理解 (3): Open vSwitch + GRE/VxLAN 组网 [Netruon Open vSwitch + GRE/VxLAN Virutal Network](13434)
4. 探索 OpenStack 之 (9) : 深入块存储服务Cinder (功能篇) (12921)
5. 理解 OpenStack + Ceph (1) : Ceph + OpenStack 集群部署和配置 (12444)

评论排行榜

1. Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network](63)
2. Neutron 理解 (14) : Neutron ML2 + Linux bridge + VxLAN 组网 (54)
3. Neutron 理解 (8): Neutron 是如何实现虚拟机防火墙的 [How Neutron Implements Security Group](34)
4. Neutron 理解 (3): Open vSwitch + GRE/VxLAN 组网 [Netruon Open vSwitch + GRE/VxLAN Virutal Network](25)
5. Neutron 理解 (5) : Neutron 是如何向 Nova 虚拟机分配固定IP地址的 (How Neutron Allocates Fixed IPs to Nova Instance) (21)

推荐排行榜

1. Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network](9)
2. 我所了解的 京东、携程、eBay、小米 的 OpenStack 云(6)
3. 理解 OpenStack 高可用 (HA) (1) : OpenStack 高可用和灾备方案 [OpenStack HA and DR](6)
4. Neutron 理解 (2): 使用 Open vSwitch + VLAN 组网 [Netruon Open vSwitch + VLAN Virutal Network](6)

- 支持半虚拟化I/O (virtio)
- 支持热插拔 (cpu, 块设备、网络设备等)
- 支持对称多处理 (Symmetric Multi-Processing, 缩写为 SMP)
- 支持实时迁移 (Live Migration)
- 支持 PCI 设备直接分配和 单根I/O 虚拟化 (SR-IOV)
- 支持 内核同页合并 (KSM)
- 支持 NUMA (Non-Uniform Memory Access, 非一致存储访问结构)

3. KVM 工具集合

- libvirt: 操作和管理KVM虚拟机的虚拟化 API, 使用 C 语言编写, 可以由 Python, Ruby, Perl, PHP, Java 等语言调用。可以操作包括 KVM, vmware, XEN, Hyper-v, LXC 等 Hypervisor。
- Virsh: 基于 libvirt 的 命令行工具 (CLI)
- Virt-Manager: 基于 libvirt 的 GUI 工具
- virt-v2v: 虚拟机格式迁移工具
- virt-* 工具: 包括 Virt-install (创建KVM虚拟机的命令行工具), Virt-viewer (连接到虚拟机屏幕的工具), Virt-clone (虚拟机克隆工具), virt-top 等
- sVirt: 安全工具

4. RedHat Linux KVM 安装

RedHat 有两款产品提供 KVM 虚拟化:

1. Red Hat Enterprise Linux: 适用于小的环境, 提供数目较少的KVM虚拟机。最新的版本包括 6.5 和 7.0.
2. Red Hat Enterprise Virtualization (RHEV): 提供企业规模的KVM虚拟化环境, 包括更简单的管理、HA, 性能优化和其它高级功能。最新的版本是 3.0.

RedHat Linux KVM:

- KVM 由 libvirt API 和基于该 API的一组工具进行管理和控制。
- KVM 支持系统资源超分, 包括内存和CPU的超分。RedHat Linux 最多支持物理 CPU 内核总数的10倍数目的虚拟CPU, 但是不支持在一个虚拟机上分配超过物理CPU内核总数的虚拟CPU。
- 支持 KSM (Kenerl Same-page Merging 内核同页合并)

RedHat Linux KVM 有如下两种安装方式:

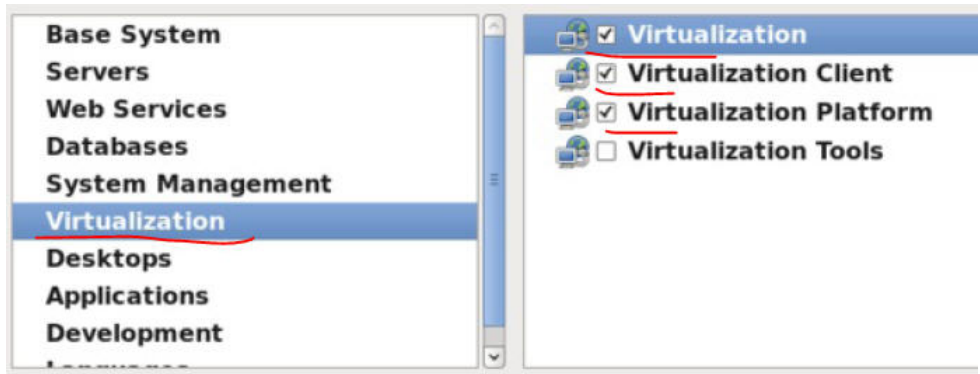
4.1 在安装 RedHat Linux 时安装 KVM

选择安装类型为 Virtualizaiton Host:

The image shows a screenshot of the Red Hat Enterprise Linux installation configuration window. Under the 'Installation type' section, 'Virtualization Host' is selected and highlighted with a red circle. Below this, there is a section for 'Please select any additional repositories that you want to use for software installation.' with checkboxes for 'High Availability', 'Load Balancer', and 'Red Hat Enterprise Linux' (which is checked). At the bottom, there is a section for 'You can further customize the software selection now, or after install via the software management application.' with radio buttons for 'Customize later' and 'Customize now' (which is selected and highlighted with a red circle).

可以选择具体的 KVM 客户端、平台和工具:

5. 理解 OpenStack 高可用 (HA)
(2) : Neutron L3 Agent HA 之虚拟路由冗余协议 (VRRP) (5)



4.2 在已有的 RedHat Linux 中安装 KVM

这种安装方式要求该系统已经被注册，否则会报错：

```
[root@rh65 ~]# yum install qemu-kvm qemu-img
Loaded plugins: product-id, refresh-packagekit, security, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Setting up Install Process
Nothing to do
```

你至少需要安装 qemu-kvm qemu-img 这两个包。

```
# yum install qemu-kvm qemu-img
```

你还可以安装其它工具包：

```
# yum install virt-manager libvirt libvirt-python python-virtinst libvirt-client
```

4.3 QEMU/KVM 代码下载编译安装

4.3.1 QEMU/KVM 的代码结构

QEMU/KVM 的代码包括几个部分：

(1) KVM 内核模块是 Linux 内核的一部分。通常 Linux 比较新的发行版 (2.6.20+) 都包含了 KVM 内核，也可以从[这里](#)得到。比如在我的 RedHat 6.5 上：

```
[root@rh65 isoimages]# uname -r
2.6.32-431.el6.x86_64
[root@rh65 isoimages]# modprobe -l | grep kvm
kernel/arch/x86/kvm/kvm.ko
kernel/arch/x86/kvm/kvm-intel.ko
kernel/arch/x86/kvm/kvm-amd.ko
```

(2) 用户空间的工具即 qemu-kvm。qemu-kvm 是 KVM 项目从 QEMU 新拉出的一个分支 ([看这篇文章](#))。在 QEMU 1.3 版本之前，QEMU 和 QEMU-KVM 是有区别的，但是从 2012 年底 GA 的 QEMU 1.3 版本开始，两者就完全一样了。

(3) Linux Guest OS virtio 驱动，也是较新的 Linux 内核的一部分了。

(4) Windows Guest OS virtio 驱动，可以从[这里](#)下载。

4.3.2 安装 QEMU

RedHat 6.5 上自带的 QEMU 太老，0.12.0 版本，最新版本都到了 2.* 了。

(1). 参考 [这篇文章](#)，将 RedHat 6.5 的 ISO 文件当作本地源

```
mount -o loop soft/rhel-server-6.4-x86_64-dvd.iso /mnt/rhel6/

vim /etc/fstab
=> /root/isoimages/soft/RHEL6.5-20131111.0-Server-x86_64-DVD1.iso /mnt/rhel6 iso9660 ro,loop

[root@rh65 qemu-2.3.0]# cat /etc/yum.repos.d/local.repo
[local]
name=local
baseurl=file:///mnt/rhel6/
enabled=1
gpgcheck=0

1 | yum clean all<br>yum update
```

(2). 安装依赖包包

```
yum install gcc
yum install autoconf
```



```
yum install autoconf automake libtool
yum install -y glib*
yum install zlib*
```

(3). 从 <http://wiki.qemu.org/Download> 下载代码, 上传到我的编译环境 RedHat 6.5.

```
tar -jzvf qemu-2.3.0.tar.bz2
cd qemu-2.3.0
./configure
make -j 4
make install
```

(4). 安装完成

```
[root@rh65 qemu-2.3.0]# /usr/local/bin/qemu-x86_64 -version
qemu-x86_64 version 2.3.0, Copyright (c) 2003-2008 Fabrice Bellard
```

(5). 为方便起见, 创建一个link

```
ln -s /usr/bin/qemu-system-x86_64 /usr/bin/qemu-kvm
```

4.3.3 安装 libvirt

可以从 [libvirt 官网](#) 下载安装包。最新的版本是 0.10.2.

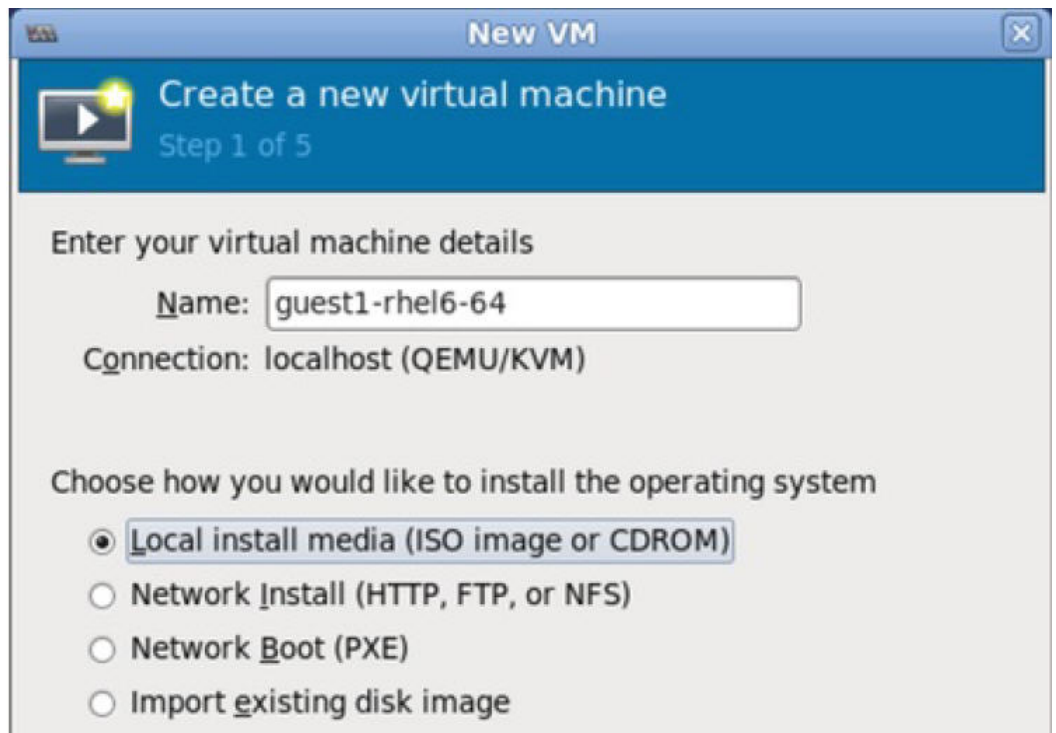
5. 创建 KVM 虚机的几种方式

5.1 使用 virt-install 命令

```
virt-install \
--name=guest1-rhel5-64 \
--file=/var/lib/libvirt/images/guest1-rhel5-64.dsk \
--file-size=8 \
--nonsparse --graphics spice \
--vcpus=2 --ram=2048 \
--location=http://example1.com/installation_tree/RHEL5.6-Serverx86_64/os \
--network bridge=br0 \
--os-type=linux \
--os-variant=rhel5.4
```

5.2 使用 virt-manager 工具





使用 VMM GUI 创建的虚机的xml 定义文件在 /etc/libvirt/qemu/ 目录中。

5.3 使用 qemu-img 和 qemu-kvm 命令行方式安装

(1) 创建一个空的qcow2格式的镜像文件

```
qemu-img create -f qcow2 windows-master.qcow2 10G
```

(2) 启动一个虚拟机，将系统安装盘挂到 cdrom，安装操作系统

```
qemu-kvm -hda windows-master.qcow2 -m 512 -boot d -cdrom  
/home/user/isos/en_winxp_pro_with_sp2.iso
```

(3) 现在你就拥有了一个带操作系统的镜像文件。你可以以它为模板创建新的镜像文件。使用模板的好处是，它会被设置为只读所以可以免于破坏。

```
qemu-img create -b windows-master.qcow2 -f qcow2 windows-clone.qcow2
```

(4) 你可以在新的镜像文件上启动虚拟机了

```
qemu-kvm -hda windows-clone.qcow2 -m 400
```

5.4 通过 OpenStack Nova 使用 libvirt API 通过编程方式来创建虚拟机（后面会介绍）

分类: [Nova,KVM](#)



 **SammyLiu**
关注 - 30
粉丝 - 470

荣誉: [推荐博客](#)
[+加关注](#)

« 上一篇: [\[转\] OpenStack Kilo 更新日志](#)


» 下一篇: [KVM 介绍 \(2\) : CPU 和内存虚拟化](#)

3
[推荐](#)

0
[反对](#)

posted on 2015-06-02 11:29 [SammyLiu](#) 阅读(9234) 评论(9) [编辑](#) [收藏](#)

评论:

#1楼 2016-03-05 17:03 | [Siffre](#) 

大神你好，小弟一直在跟着你的步伐前进，请多多指点，这是小弟的博客，有时间来多多指点，小弟感激不尽！ 博客地址：lanzhihi.org
[支持\(0\)](#) [反对\(0\)](#)

#2楼[楼主] 2016-03-05 17:51 | [SammyLiu](#) 

@ [Siffre](#)

实在不敢当大神，就是喜欢随手记点东西，希望在帮助自己的同时对别人也有些帮助或者启发。我看了你的博客，版面和内容都挺不错，加

油，贵在坚持！

支持(0) 反对(0)

#3楼 2016-03-05 17:52 | Siffre

@ SammyLiu

实时在线a，加个微信吧，ctym361

支持(0) 反对(0)

#4楼[楼主] 2016-03-05 17:53 | SammyLiu

@ Siffre

已加

支持(0) 反对(0)

#5楼 2016-08-18 15:47 | cloudcloud

大大讲的太好啦

支持(0) 反对(0)

#6楼 2017-02-20 13:42 | edward_lii

你的文章写得很好，但我有些疑问，能告知一下你的联系方式吗？

支持(0) 反对(0)

#7楼[楼主] 2017-02-27 09:05 | SammyLiu

@ edward_lii

见站内信

支持(0) 反对(0)

#8楼 2017-03-27 09:42 | 不见天高

楼主你好，可以加个微信请教你不，z157984031

支持(0) 反对(0)

#9楼 2017-04-05 10:35 | 大相林

挺好，知识面非常全面，很深入。

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
- 【报表】Excel 报表开发18 招式，人人都能做报表
- 【活动】阿里云海外云服务全面降价助力企业全球布局
- 【实用】40+篇云服务器操作及运维基础知识！



- 最新IT新闻:
- 知乎上线视频功能，以后看教程更方便了
 - 一年只赚2万元：乐视游戏或被出售
 - Unity获得4亿美元投资，现估值为26亿美元
 - 直播对陌陌的意义，就像王者荣耀之于腾讯游戏
 - 死磕支付宝？苏宁金融发布“星辰计划”：扫码支付返888元
- » [更多新闻...](#)

最新知识库文章:

- [程序员的工作、学习与绩效](#)
- [软件开发为什么很难](#)
- [唱吧DevOps的落地，微服务CI/CD的范本技术解读](#)
- [程序员，如何从平庸走向理想？](#)
- [我为什么鼓励工程师写blog](#)
- » [更多知识库文章...](#)

Powered by: [博客园](#) 模板提供: [沪江博客](#) Copyright ©2017 SammyLiu