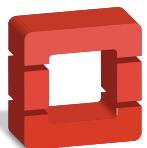


# OpenStack

## 虚拟机镜像制作指南

current (June 29, 2015)



openstack™

# OpenStack 虚拟机镜像制作指南

current (2015-06-29)

版权 © 2013-2015 OpenStack基金会 Some rights reserved.

## 摘要

这个指南描述了如何获取，创建以及修改 OpenStack 兼容的虚拟机镜像。



Except where otherwise noted, this document is licensed under  
Creative Commons Attribution 3.0 License.  
<http://creativecommons.org/licenses/by/3.0/legalcode>

# 目录

前言 .....	vi
约定 .....	vi
文档变更历史 .....	vi
1. 介绍 .....	1
镜像的磁盘格式和镜像的容器格式 .....	2
镜像元数据 .....	3
2. 获取镜像 .....	5
CentOS images .....	5
CirrOS (测试镜像) .....	5
Ubuntu 官方镜像 .....	5
红帽企业Linux官方镜像 .....	6
官方Fedora 镜像 .....	6
官方 openSUSE 和 SLES 镜像 .....	6
官方 Debian 镜像 .....	6
其他Linux发行版的官方镜像 .....	7
Rackspace 云计算镜像 (多个发行版) .....	7
微软 Windows 镜像 .....	7
3. OpenStack Linux 镜像需求 .....	8
磁盘分区以及在启动时调整根分区大小 (cloud-init) .....	8
无硬编码的 MAC 地址信息 .....	10
确保 ssh 服务在运行 .....	10
禁用防火墙 .....	10
使用 SSH 公钥 (cloud-init) 访问虚拟机 .....	11
处理用户数据以及其他元数据 (cloud-init) .....	12
确认镜像输出启动日志到控制台 .....	12
内核内对半虚拟化的 Xen 支持 (仅Xen 虚拟化) .....	12
管理镜像缓存 .....	13
4. 更改镜像内容 .....	14
guestfish .....	14
guestmount .....	15
virt-* tools .....	16
loop(环回)设备 , kpartx, 网络块设备 .....	17
5. 手动创建镜像 .....	20
检查 libvirt 默认网络是否运行 .....	20
使用 virt-manager X11 图形界面 .....	20
使用 virt-install 且使用本地 VNC 客户端连接 .....	21
示例 : CentOS 镜像 .....	22
示例 : Ubuntu 镜像 .....	29
示例 : Fedora 镜像 .....	35
示例 : 微软Windows镜像 .....	38
示例 : FreeBSD 镜像 .....	39
6. 支持创建镜像的工具 .....	43
Diskimage-builder .....	43
0z .....	43
VMBuilder .....	45
BoxGrinder .....	45
VeeWee .....	45
Packer .....	45

imagefactory .....	45
SUSE Studio .....	45
7. 转换镜像格式 .....	46
A. 社区支持 .....	47
文档 .....	47
问答论坛 .....	48
OpenStack 邮件列表 .....	48
OpenStack 维基百科 .....	48
Launchpad的Bug区 .....	48
The OpenStack 在线聊天室频道 .....	49
文档反馈 .....	50
OpenStack分发包 .....	50

## 表格清单

3.1. 镜像缓存管理配置选项 .....	13
7.1. qemu-img 格式字符列表 .....	46

# 前言

约定 .....	vi
文档变更历史 .....	vi

## 约定

OpenStack文档使用了多种排版约定。

## 声明

以下形式的标志为注意项



### 注意

便签或提醒



### 重要

提醒用户进一步操作之前必须谨慎



### 警告

关于有丢失数据的风险或安全漏洞的危险信息提示

## 命令行提示符

\$ 提示符 \$ 提示符表示任意用户，包括 root 用户，都可以运行的命令。

# 提示符 # 提示符表明命令仅为 root 可以执行。当然，用户如果可以使用 sudo 命令的话，也可以运行此提示符后面的命令。

## 文档变更历史

此版本的向导完全取代旧版本的，且所有旧版本的在此版本中不再生效。

下面表格描述的是近期的改动：

Revision Date	Summary of Changes
October 15, 2014	· Juno 发行，次要更新
April 17, 2014	· Icehouse 发行，次要更新 - 移动特性列表到 Command-Line Interface Reference 并添加 Windows 时区相关。
October 25, 2013	· 添加镜像格式，属性相关信息。
October 17, 2013	· Havana 发行版。
June 4, 2013	· 为一致性更新标题。
May 28, 2013	· 本指南初始版本。

# 第 1 章 介绍

镜像的磁盘格式和镜像的容器格式 .....	2
镜像元数据 .....	3

在你拥有虚拟机镜像（也有人叫”虚拟器件”）之前，你的 OpenStack 计算云不太有用，这个指南描述了如何获取，创建以及修改 OpenStack 兼容的虚拟机镜像。

为了简化，有时文档使用“镜像”代替“虚拟机镜像”。

虚拟机镜像是什么？

一个虚拟机镜像是一个文件，文件内包含了已经安装好可启动操作系统的虚拟磁盘。

Virtual machine images come in different formats, some of which are described below.

Raw “raw” 镜像格式是最最简单的，并且是被 KVM 和 Xen 原生支持的格式，你可以想象裸格式镜像和块设备文件是二进制位相当的，就好像从块设备拷贝过来的，比方说，使用 dd 命令将 /dev/sda 内容拷贝到一个文件。

qcow2 The [qcow2](#) (QEMU copy-on-write version 2) format is commonly used with the KVM hypervisor. It has some additional features over the raw format, such as:

- Using sparse representation, so the image size is smaller.
- Support for snapshots.

因为 qcow2 镜像是稀疏的，因此，qcow2 镜像通常比裸格式镜像小，更小的文件意味着上传更快，因此通常转换裸格式镜像到 qcow2 格式上传比直接上传裸格式镜像文件更快。



## 注意

因为裸格式镜像不支持快照，OpenStack 计算节点在需要的时候将会自动转换裸格式镜像到 qcow2 格式。

AMI/AKI/ARI The [AMI/AKI/ARI](#) format was the initial image format supported by Amazon EC2. The image consists of three files:

- AMI (亚马逊虚机镜像):

裸格式镜像，如前面描述。

- AKI (亚马逊内核镜像)

虚拟化启动镜像时最先加载的内核文件。对 Linux 机器，对应的应该 是 vmlinuz 文件。

- ARI (亚马逊 Ramdisk 镜像)

系统启动时可选的 ramdisk 文件，对 Linux 机器，对应的是 initrd 文件。

UEC tarball	UEC (Ubuntu 企业云) 文件是一个包含了 AMI AKI ARI文件的tar包压缩文件。
VMDK	VMware's ESXi hypervisor uses the <a href="#">VMDK</a> (Virtual Machine Disk) format for images.
VDI	VirtualBox uses the <a href="#">VDI</a> (Virtual Disk Image) format for image files. None of the OpenStack Compute hypervisors support VDI directly, so you will need to convert these files to a different format to use them with OpenStack.
VHD	微软 Hyper-V 使用 VHD 镜像格式。
VHDX	The version of Hyper-V that ships with Microsoft Server 2012 uses the newer <a href="#">VHDX</a> format, which has some additional features over VHD such as support for larger disk sizes and protection against data corruption during power failures.
OVF	<a href="#">OVF</a> (Open Virtualization Format) is a packaging format for virtual machines, defined by the Distributed Management Task Force (DMTF) standards group. An OVF package contains one or more image files, a .ovf XML metadata file that contains information about the virtual machine, and possibly other files as well.  OVF 包可通过几种方式分发，例如，可以通过一系列分散的文件分发或者通过一个tar归档文件 (.ova后缀) 分发。  OpenStack 计算节点目前并不支持 OVF 包，如果你想要在Openstack中使用，你需要从OVF中释放磁盘镜像文件。
ISO	The <a href="#">ISO</a> format is a disk image formatted with the read-only ISO 9660 (also known as ECMA-119) filesystem commonly used for CDs and DVDs. While we don't normally think of ISO as a virtual machine image format, since ISOs contain bootable filesystems with an installed operating system, you can treat them the same as you treat other virtual machine image files.

## 镜像的磁盘格式和镜像的容器格式

When you add an image to the Image service, you can specify its disk and container formats.

### 磁盘格式

虚拟机的磁盘格式指的是底层的磁盘镜像格式，虚拟设备厂商有不同的格式来放置虚拟机磁盘镜像包含的信息。

设置你的镜像磁盘格式为下面的值：

- raw: An unstructured disk image format; if you have a file without an extension it is possibly a raw format.
- vhd: The VHD disk format, a common disk format used by virtual machine monitors from VMware, Xen, Microsoft, VirtualBox, and others.

- vmdk: Common disk format supported by many common virtual machine monitors.
- vdi: Supported by VirtualBox virtual machine monitor and the QEMU emulator.
- iso: An archive format for the data contents of an optical disc, such as CD-ROM.
- qcow2: Supported by the QEMU emulator that can expand dynamically and supports Copy on Write.
- aki: An Amazon kernel image.
- ari: An Amazon ramdisk image.
- ami: An Amazon machine image.

## 容器格式

镜像容器格式表示包含虚拟机镜像的文件是否也包含虚拟机的metadata数据。



### 注意

The Image service and other OpenStack projects do not currently support the container format. It is safe to specify bare as the container format if you are unsure.

你可设置你的镜像容器格式为以下之一：

- bare. 镜像没有容器或者没有封装metadata数据。
- ovf, OVF容器格式。
- aki, Amazon 内核镜像。
- ari, Amazon ramdisk 镜像。
- ami, Amazon 虚机镜像。

## 镜像元数据

Image metadata can help end users determine the nature of an image, and is used by associated OpenStack components and drivers which interface with the Image service.

元数据还能探测知主机的调度。如果设置了镜像的 property 选项，而且计算节点也配置了启用（默认的）ImagePropertiesFilter 调度过滤器。那么调度器在调度时只需要考虑哪些符合属性需求的计算节点。



### 注意

计算节点的 ImagePropertiesFilter 值在 /etc/nova/nova.conf 文件的 scheduler\_default\_filter 参数设定。

You can add metadata to Image service images by using the --property key=value parameter with the glance image-create or glance image-update command. More than one property can be specified. For example:

```
$ glance image-update img-uuid --property architecture=arm --property hypervisor_type=qemu
```

Common image properties are also specified in the /etc/glance/schema-image.json file. For a complete list of valid property keys and values, refer to the [OpenStack Command-Line Reference](#).

镜像的所有相关联的属性可以使用 glance image-show 命令显示。例如：

Property	Value
Property 'base_image_ref'	397e713c-b95b-4186-ad46-6126863ea0a9
Property 'image_location'	snapshot
Property 'image_state'	available
Property 'image_type'	snapshot
Property 'instance_type_ephemeral_gb'	0
Property 'instance_type_flavorid'	2
Property 'instance_type_id'	5
Property 'instance_type_memory_mb'	2048
Property 'instance_type_name'	m1.small
Property 'instance_type_root_gb'	20
Property 'instance_type_rxtx_factor'	1
Property 'instance_type_swap'	0
Property 'instance_type_vcpu_weight'	None
Property 'instance_type_vcpus'	1
Property 'instance_uuid'	84c6e57d-a6b1-44b6-81eb-fcb36af31b5
Property 'kernel_id'	df430cc2-3406-4061-b635-a51c16e488ac
Property 'owner_id'	66265572db174a7aa66eba661f58eb9e
Property 'ramdisk_id'	3cf852bd-2332-48f4-9ae4-7d926d50945e
Property 'user_id'	376744b5910b4b4da7d8e6cb483b06a8
checksum	8e4838effa1969ad591655d6485c7ba8
container_format	ami
created_at	2013-07-22T19:45:58
deleted	False
disk_format	ami
id	7e5142af-1253-4634-bcc6-89482c5f2e8a
is_public	False
min_disk	0
min_ram	0
name	myCirrosImage
owner	66265572db174a7aa66eba661f58eb9e
protected	False
size	14221312
status	active
updated_at	2013-07-22T19:46:42



## 镜像转换的卷属性

当从镜像创建块存储卷的时候，需要考虑镜像的属性。如果你修改镜像的核心属性，你也应该更新块存储配置。修改所有匹配你设置在镜像服务的核心属性的控制节点的/etc/cinder/cinder.conf 文件的 glance\_core\_properties 值。

# 第 2 章 获取镜像

CentOS images .....	5
CirrOS (测试镜像) .....	5
Ubuntu 官方镜像 .....	5
红帽企业Linux官方镜像 .....	6
官方Fedora 镜像 .....	6
官方 openSUSE 和 SLES 镜像 .....	6
官方 Debian 镜像 .....	6
其他Linux发行版的官方镜像 .....	7
Rackspace 云计算镜像 (多个发行版) .....	7
微软 Windows 镜像 .....	7

获取 OpenStack 兼容的虚拟机镜像的最简单的方法就是直接下载已经制作好的镜像。

## CentOS images

The CentOS project maintains official images for direct download.

- [CentOS 6 images](#)
- [CentOS 7 images](#)

## CirrOS (测试镜像)

CirrOS is a minimal Linux distribution that was designed for use as a test image on clouds such as OpenStack Compute. You can download a CirrOS image in various formats from the [CirrOS download page](#).

If your deployment uses QEMU or KVM, we recommend using the images in qcow2 format. The most recent 64-bit qcow2 image as of this writing is [cirros-0.3.3-x86\\_64-disk.img](#).



### 注意

CirrOS 镜像的登陆账号是 cirros，密码是 cubswin:)

## Ubuntu 官方镜像

Canonical maintains an [official set of Ubuntu-based images](#).

Images are arranged by Ubuntu release, and by image release date, with "current" being the most recent. For example, the page that contains the most recently built image for Ubuntu 14.04 "Trusty Tahr" is <http://cloud-images.ubuntu.com/trusty/current/>. Scroll to the bottom of the page for links to images that can be downloaded directly.

If your deployment uses QEMU or KVM, we recommend using the images in qcow2 format. The most recent version of the 64-bit QCOW2 image for Ubuntu 14.04 is [trusty-server-cloudimg-amd64-disk1.img](#).



## 注意

Ubuntu 云镜像的登陆账号是 ubuntu。

# 红帽企业ILinux官方镜像

Red Hat maintains official Red Hat Enterprise Linux cloud images. A valid Red Hat Enterprise Linux subscription is required to download these images.

- [Red Hat Enterprise Linux 7 KVM Guest Image](#)
- [Red Hat Enterprise Linux 6 KVM Guest Image](#)



## 注意

在红帽企业Linux镜像中，登陆账户是 cloud-user。

# 官方Fedora 镜像

The Fedora project maintains a list of official cloud images at <https://getfedora.org/en/cloud/download/>. The images include the cloud-init utility to support key and user data injection. The default user name is fedora.



## 注意

在Fedora 镜像中，登陆账户是 fedora。

# 官方 openSUSE 和 SLES 镜像

SUSE provides images for [openSUSE](#). For SUSE Linux Enterprise Server (SLES), custom images can be built with a web-based tool called [SUSE Studio](#). SUSE Studio can also be used to build custom openSUSE images.

# 官方 Debian 镜像

Since January 2015, [Debian](#) provides images for direct download. They are now made at the same time as the CD and DVD images of Debian. However, until Debian 8.0 (aka Jessie) is out, these images are the weekly built images of the testing distribution.

如果你想创建 Debian 7.0 镜像（也就是 Wheezy，Debian 当前稳定版本）。你可以使用用于创建官方的 Debian 镜像的 `openstack-debian-images` 软件包，它提供既简单的脚本来创建。这个包在 Debian 未稳定分支 Debian Jessie 内，通过使用 `wheezy-backports` 源。要生成 Wheezy 镜像，仅需运行：

```
# build-openstack-debian-image -r wheezy
```

If building the image for Wheezy, packages like `cloud-init`, `cloud-utils` or `cloud-initramfs-growroot` will be pulled from `wheezy-backports`. Also, the current version of `bootlogd` in Wheezy doesn't support logging to multiple consoles, which is needed so that both the OpenStack Dashboard console and the `nova console-log` console works. However, a `fixed`

version is available from the non-official [GPLHost repository](#). To install it on top of the image, it is possible to use the --hook-script option of the build-openstack-debian-image script, with this kind of script as parameter:

```
#!/bin/sh  
  
cp bootlogd_2.88dsf-41+deb7u2_amd64.deb ${BODI_CHROOT_PATH}  
chroot ${BODI_CHROOT_PATH} dpkg -i bootlogd_2.88dsf-41+deb7u2_amd64.deb  
rm ${BODI_CHROOT_PATH}/bootlogd_2.88dsf-41+deb7u2_amd64.deb
```



### 注意

In a Debian image, the login account is admin.

## 其他Linux发行版的官方镜像

在本文档编写时，我们并未发现其他Linux发行版提供镜像下载。

## Rackspace 云计算镜像（多个发行版）

Rackspace Cloud Builders maintains a list of pre-built images from various distributions (Red Hat, CentOS, Fedora, Ubuntu). Links to these images can be found at [rackerjoe/oz-image-build on GitHub](#).

## 微软 Windows 镜像

Cloudbase Solutions hosts an OpenStack Windows Server 2012 Standard Evaluation image that runs on Hyper-V, KVM, and XenServer/XCP.

# 第 3 章 OpenStack Linux 镜像需求

磁盘分区以及在启动时调整根分区大小 (cloud-init) .....	8
无硬编码的 MAC 地址信息 .....	10
确保 ssh 服务在运行 .....	10
禁用防火墙 .....	10
使用 SSH 公钥 (cloud-init) 访问虚拟机 .....	11
处理用户数据以及其他元数据 (cloud-init) .....	12
确认镜像输出启动日志到控制台 .....	12
内核内对半虚拟化的 Xen 支持 (仅Xen 虚拟化) .....	12
管理镜像缓存 .....	13

For a Linux-based image to have full functionality in an OpenStack Compute cloud, there are a few requirements. For some of these, you can fulfill the requirements by installing the [cloud-init](#) package. Read this section before you create your own image to be sure that the image supports the OpenStack features that you plan to use.

- 磁盘分区以及在启动时调整根分区大小 (cloud-init)
- 无硬编码的 MAC 地址信息
- 启用SSH服务
- 禁用防火墙
- 使用SSH 公钥访问虚拟机 (cloud-init)
- 处理用户以及其他的数据 (cloud-init)
- Linux 内核对半虚拟化 Xen的支持 (Xen hypervisor only with Linux kernel version < 3.0)

## 磁盘分区以及在启动时调整根分区大小 (cloud-init)

当创建 Linux 镜像时，你必须决定如何分区。选择不同的分区方法会对重新调整分区大小有不同的影响，请看以下章节的描述。

镜像磁盘的大小在最开始创建磁盘的时候就已经决定了的，然而，OpenStack 通过指定不同的硬件配置规格让你启动磁盘大小不同的虚拟机。例如，如果你的镜像是5GB，你创建的虚拟机指定的虚拟机规格是 m1.small。最终创建的虚拟机的系统磁盘是10GB。当虚拟机的磁盘扩充的时候，在磁盘的后面以0填充。

你的镜像必须能在启动的时候重新伸缩大小以便匹配最终用户的需求。否则，在虚拟机启动后，你需要手工改变分区的容量才能使用虚拟机指定的硬件配置规格的磁盘分容量大于你镜像原有容量的那部分容量。

### Xen: 单一 ext3/ext4 分区 ( 无LVM , 无 /boot ,无swap分区区 )

如果你使用 OpenStack XenAPI 驱动，那么计算服务在虚拟机启动时会自动调整分区和文件系统大小。在满足以下条件的时候会自动调整容量需：

- 在镜像的属性设置了 auto\_disk\_config=True 。

- 镜像的磁盘只有一个分区。
- 仅有的分区文件系统是 ext3 或者 ext4.

因此，如果你使用 Xen，我们推荐当你创建镜像时，只建立一个 ext3 或者 ext4 分区（而且没有LVM管理）。否则，继续往下看。

## 非 Xen 平台安装 cloud-init/cloud-tools : 单 ext3/ext4 分区 (无 LVM, 无 /boot, 无 SWAP)

你的镜像必须满足以下配置：

- The partition table for the image describes the original size of the image.
- The file system for the image fills the original size of the image.

然后，在启动过程中，你必须：

- 修改分区表识别磁盘额外的空间。
  - 如果你没使用 LVM,你需要修改分区表扩充已经存在的根分区容纳磁盘额外的空间。
  - 如果你使用了LVM，你可以添加新 LVM 条目到分区表，创建新 LVM 物理卷，然后添加 LVM 物理卷到 卷组，然后扩充 root 卷的容量。
- 重改变 root 卷文件系统。

实现以上操作最简单的方法是安装以下软件到镜像内：

- [cloud-utils](#) package, which contains the growpart tool for extending partitions.
- [cloud-initramfs-growroot](#) package for Ubuntu, Debian and Fedora, which supports resizing root partition on the first boot.
- Centos 和 RHEL 上使用[cloud-initramfs-growroot](#)
- [cloud-init](#) package.

通过安装这些包，镜像在启动时完成root分区改变大小。例如，通过设置/etc/rc.local 文件，这些包包在Ubuntu 和 Debian 包管理库内以及 EPEL 库（ Fedora/RHEL/CentOS/ Scientific 系列 Linux）

If you cannot install cloud-initramfs-tools, Robert Plestenjak has a GitHub project called [linux-rootfs-resize](#) that contains scripts that update a ramdisk by using growpart so that the image resizes properly on boot.

如果你能安装 cloud-utils 和 cloud-init包，我们推荐你创建镜像时，只创建一个 ext3 和 ext4 分区（且不使用LVM管理）

## 非 Xen 环境且无 cloud-init/cloud-tools软件包 : 使用 LVM

如果你不能安装 cloud-init 和 cloud-tools 软件包到镜像内但你还想镜像支持伸缩大小，你必须编写脚本让你的镜像在启动的时候修改分区表。在这种情况下，我们推荐使用 LVM 来管

理你的磁盘分区。由于 Linux 内核限制（本文档编写时），当磁盘有分区被挂载的时候你不能修改磁盘分区表，但是你可以使用 LVM。

你的脚本需要完成以下功能：

1. 检查磁盘额外可用空间。例如，解析 `parted /dev/sda --script "print free"` 的输出。
2. 在额外的空间上创建 LVM 分区。例如：`parted /dev/sda --script "mkpart lvm ..."`。
3. 创建新的物理卷(PV)。例如：`pvcreate /dev/sda6`。
4. 使用这个屋里分区扩展卷组 (VG)。例如：`vgextend vg00 /dev/sda6`。
5. 使用分区所有空间扩展包含根分区的逻辑卷 (LV)。例如，`lvextend /dev/mapper/node-root /dev/sda6`。
6. 重新改变根文件系统大小。例如，`resize2fs /dev/mapper/node-root`。

你无须 `/boot` 分区，除非你的镜像使用的是较老的 Linux 发行版，那些发行版要求 `/boot` 目录不能被 LVM 管理。

## 无硬编码的 MAC 地址信息

You must remove the network persistence rules in the image because they cause the network interface in the instance to come up as an interface other than eth0. This is because your image has a record of the MAC address of the network interface card when it was first installed, and this MAC address is different each time the instance boots. You should alter the following files:

- Replace `/etc/udev/rules.d/70-persistent-net.rules` with an empty file (contains network persistence rules, including MAC address).
- Replace `/lib/udev/rules.d/75-persistent-net-generator.rules` with an empty file (this generates the file above).
- Remove the HWADDR line from `/etc/sysconfig/network-scripts/ifcfg-eth0` on Fedora-based images.



### 注意

如果你删除网络持久规则文件，系统启动时你会看到内核警告信息，这是为什么我们推荐使用空文件替代的原因。

## 确保 ssh 服务在运行

在镜像内你必须安装 SSH 服务并且确保它在启动时自动启动，否则你不能通过 ssh 连接到你在 OpenStack 内的虚拟机。ssh 服务包名通常叫做 `openssh-server`。

## 禁用防火墙

通常，我们推荐你关闭掉镜像内所有的防火墙，使用 OpenStack 安全组来限制访问你的虚拟机。因为在镜像内开启防火墙会增加在不能访问虚拟机的时候诊断问题时的难度。

# 使用 SSH 公钥（cloud-init）访问虚拟机

访问在 OpenStack 内运行的虚拟机典型的方法是通过 ssh 使用公钥认证。为了实现这个，你的虚拟机镜像必须被配置为在虚拟机启动时能从 OpenStack 元数据服务获取ssh 公钥或者使用config 磁盘。

## 使用 cloud-init 获取公钥

cloud-init 软件从元数据服务自动获取公钥并存放在账户内。不同发行版的账户不一样。基于 Ubuntu 系的虚拟机，账户是 ubuntu 。Fedora 系的虚拟机，账户是 ec2-user 。

你可以编辑 /etc/cloud/cloud.cfg 文件添加一行修改 cloud-init 使用的账户名，例如，添加以下这行到配置文件，来配置 cloud-init 存放公钥到admin账户：

```
user: admin
```

## 编写脚本获取公钥

如果你没有或者不希望安装 cloud-init包到虚拟机里，你可以自己编写脚本来获取公钥并存放到用户账户内。

To fetch the ssh public key and add it to the root account, edit the /etc/rc.local file and add the following lines before the line "touch /var/lock/subsys/local". This code fragment is taken from the [rackerjoe oz-image-build CentOS 6 template](#).

```
if [ ! -d /root/.ssh ]; then
    mkdir -p /root/.ssh
    chmod 700 /root/.ssh
fi

# Fetch public key using HTTP
ATTEMPTS=30
FAILED=0
while [ ! -f /root/.ssh/authorized_keys ]; do
    curl -f http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key > /tmp/metadata-key 2>/dev/null
    if [ $? -eq 0 ]; then
        cat /tmp/metadata-key >> /root/.ssh/authorized_keys
        chmod 0600 /root/.ssh/authorized_keys
        restorecon /root/.ssh/authorized_keys
        rm -f /tmp/metadata-key
        echo "Successfully retrieved public key from instance metadata"
        echo "*****"
        echo "AUTHORIZED KEYS"
        echo "*****"
        cat /root/.ssh/authorized_keys
        echo "*****"
    else
        FAILED=`expr $FAILED + 1`
        if [ $FAILED -ge $ATTEMPTS ]; then
            echo "Failed to retrieve public key from instance metadata after $FAILED attempts, quitting"
            break
        fi
        echo "Could not retrieve public key from instance metadata (attempt #$FAILED/$ATTEMPTS), retrying in 5 seconds..."
        sleep 5
    fi
done
```

fi  
done



## 注意

有些 VNC 客户会使用分号 (;) 替换冒号 (:) 以及用中划线 (-) 替换下划线 (\_) ，确认你输入的是 http: 而不是 http; ， authorized\_keys 而不是 authorized-keys 。

## 处理用户数据以及其他元数据 (cloud-init)

In addition to the ssh public key, an image might need additional information from OpenStack, such as [Provide user data to instances](#), that the user submitted when requesting the image. For example, you might want to set the host name of the instance when it is booted. Or, you might wish to configure your image so that it executes user data content as a script on boot.

You can access this information through the metadata service or referring to [Store metadata on the configuration drive](#). As the OpenStack metadata service is compatible with version 2009-04-04 of the Amazon EC2 metadata service, consult the Amazon EC2 documentation on [Using Instance Metadata](#) for details on how to retrieve user data.

支持这个功能的最简单的办法是安装 cloud-init 软件包到你的镜像，它默认配置为将用户数据视为可执行脚本并且设置主机名。

## 确认镜像输出启动日志到控制台

你必须配置镜像以便镜像内核输出启动日志到 ttyS0 设备，尤其是 console=ttyS0 参数必须在启动时传递给内核。

如果你的镜像使用 grub2 做为引导程序，在配置文件内应该有一行配置内容。例如， /boot/grub/grub.cfg，看起来应该如下：

```
linux /boot/vmlinuz-3.2.0-49-virtual root=UUID=6d2231e4-0975-4f35-a94f-56738c1a8150 ro console=ttyS0
```

如果 console=ttyS0 不存在，你需要修改 grub 配置，通常，你不应该直接更新 grub.cfg 文件，因为它是程序自动生成的。作为替代的，你应该编辑 /etc/default/grub 文件并修改 GRUB\_CMDLINE\_LINUX\_DEFAULT 的值：

```
GRUB_CMDLINE_LINUX_DEFAULT="console=ttyS0"
```

然后，更新 grub 配置，在 Debian 系列的系统如 Ubuntu，运行以下命令：

```
# update-grub
```

在 Fedora 系列的系统，如 RHEL 和 CentOS，以及 openSUSE，运行以下命令：

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

## 内核内对半虚拟化的 Xen 支持（仅Xen 虚拟化）

在 Linux 内核 3.0以前，内核的主线分支不支持 Xen 半虚拟化虚拟机（即 Xen DomU 虚拟机）。如果你运行 Xen 半虚拟化，并且你想创建内核在3.0版本之前的虚拟机镜像，你不许确保镜像启动内核已经编译了 Xen 支持。

## 管理镜像缓存

使用 nova.conf 的选项来控制在 /var/lib/nova/instances/\_base/ 目录中是否保存或者多长时间，未使用的基础镜像。如果你配置了虚拟机活迁移。所有的计算节点还需要共享同一个 /var/lib/nova/instances/ 目录

For information about libvirt images in OpenStack, see [The life of an OpenStack libvirt image from Pádraig Brady](#).

**表 3.1. 镜像缓存管理配置选项**

配置属性=默认值	(类型) 描述
preallocate_images=None	(字符串型) 虚拟机镜像预分配模式： · none。不分配存储。 · space. Storage is fully allocated at instance start. The \$instance_dir/images are <b>fallocated</b> to immediately determine if enough space is available, and to possibly improve VM I/O performance due to ongoing allocation avoidance, and better locality of block allocations.
remove_unused_base_images=True	(布尔型) 未使用的基础镜像是否被删除？当设置为 True 的时候，删除未使用的基础镜像的时间间隔在下面两个参数设置。如果设置为 False，基础镜像将一直不被删除。
remove_unused_original_minimum_age_seconds=86400	(整数型)，未使用以及未修改大小的镜像，低于这个值，不删除。默认是86400秒，或24小时。
remove_unused_resized_minimum_age_seconds=3600	(整数型) 未使用且大小改变了的基础镜像镜像文件，低于这个值的不删除。默认是3600秒，或1小时。

查看以上设置对被删除的虚拟机镜像文件的影响，查看以下目录：

```
# ls -lash /var/lib/nova/instances/_base/
```

在 /var/log/compute/compute.log 文件，查看以下部分日志。

```
2012-02-18 04:24:17 41389 WARNING nova.virt.libvirt.imagecache [-] Unknown base file: /var/lib/nova/instances/_base/06a057b9c7b0b27e3b496f53d1e88810a0d1d5d3_20
2012-02-18 04:24:17 41389 INFO nova.virt.libvirt.imagecache [-] Removable base files: /var/lib/nova/instances/_base/06a057b9c7b0b27e3b496f53d1e88810a0d1d5d3_20
2012-02-18 04:24:17 41389 INFO nova.virt.libvirt.imagecache [-] Removing base file: /var/lib/nova/instances/_base/06a057b9c7b0b27e3b496f53d1e88810a0d1d5d3
```

由于 remove\_unused\_original\_minimum\_age\_seconds 的值默认是86400秒（24小时），你可以等待这个时间间隔来查看基础镜像是否被删除，或者在 nova.conf 内设置更短的时间间隔。修改 nova.conf 后，重启所有的 nova 服务。

# 第4章 更改镜像内容

guestfish .....	14
guestmount .....	15
virt-* tools .....	16
loop(环回)设备，kpartx,网络块设备 .....	17

Once you have obtained a virtual machine image, you may want to make some changes to it before uploading it to the OpenStack Image service. Here we describe several tools available that allow you to modify images.



## 警告

不要使用这些工具修改正在运行的虚拟机镜像，这些工具被设计成只能修改未运行的虚拟机镜像。

## guestfish

The guestfish program is a tool from the [libguestfs](#) project that allows you to modify the files inside of a virtual machine image.



## 注意

guestfish并不直接mount镜像文件到本地文件系统，而是提供一个shell接口，你可以通过这个shell接口对镜像内文件做查看，编辑，删除操作，诸如 touch, chmod, 和 rm的 guestfish 命令，就像普通bash命令一样。

## guestfish 修改示例：

有时，你必须删除虚拟机镜像内任何包含虚拟机安装时分配给虚拟机网卡的MAC的文件，因为用镜像文件创建新虚拟机时网卡MAC地址和创建镜像的虚拟机并不一样。这个示例通过删除 /etc/udev/rules.d/70-persistent-net.rules文件以及删除/etc/sysconfig/network-scripts/ifcfg-eth0文件中HWADDR一行来演示了如何使用guestfish删除镜像中和原MAC地址相关联的地方。

假设你有一个文件名为centos63\_desktop.img的 CentOS qcow2 格式的虚拟机镜像。用root用户挂载这个镜像为可读可写模式，如下：

```
# guestfish --rw -a centos63_desktop.img

Welcome to guestfish, the libguestfs filesystem interactive shell for
editing virtual machine filesystems.

Type: 'help' for help on commands
      'man' to read the manual
      'quit' to quit the shell

>fs>
```

通过上面命令，进入了 guestfish 会话，注意 guestfish提示符像一条鱼的样子：><fs>。

在做任何操作之前，必须先在 guestfish提示符运行run命令。它会启动一个虚拟机，用于完成我们所有的对文件的操作。

```
>fs> run
```

通过list-filesystems命令，我们可查看镜像内的文件系统列表：

```
>fs> list-filesystems  
/dev/vda1: ext4  
/dev/vg_centosbase/lv_root: ext4  
/dev/vg_centosbase/lv_swap: swap
```

我们必须挂载包含根分区的那个逻辑卷：

```
>fs> mount /dev/vg_centosbase/lv_root /
```

下一步，我们想删除一个文件，使用guestfish命令 rm，就像使用普通shell命令一样。

```
>fs> rm /etc/udev/rules.d/70-persistent-net.rules
```

我们想编辑 ifcfg-eth0 文件以便删除 HWADDR 这一行。edit命令将文件拷贝文件到主机，调用编辑器打开文件，编辑完后将它又拷贝回原来的位置。

```
>fs> edit /etc/sysconfig/network-scripts/ifcfg-eth0
```

如果你想修改镜像让他在启动时加载8021q内核模块，你必须在/etc/sysconfig/modules/目录下简历一个可执行脚本。你可以使用touch命令创建一个空文件，然后使用edit命令编辑它，然后使用 chmod 命令让它具有可执行权限。

```
>fs> touch /etc/sysconfig/modules/8021q.modules  
>fs> edit /etc/sysconfig/modules/8021q.modules
```

我们添加以下一行到文件内，然后保存：

```
modprobe 8021q
```

然后设置文件可执行权限：

```
>fs> chmod 0755 /etc/sysconfig/modules/8021q.modules
```

一切都完成后，我们可以使用 exit命令退出guestfish提示符：

```
>fs> exit
```

## guestfish更多功能

There is an enormous amount of functionality in guestfish and a full treatment is beyond the scope of this document. Instead, we recommend that you read the [guestfs-recipes](#) documentation page for a sense of what is possible with these tools.

## guestmount

对于某些类型的修改，你可以使用更简单的方式直接挂载镜像内文件系统。guestmount 程序，同样出自于 libguestfs 项目。

例如，挂载 centos63\_desktop.qcow2镜像内的根分区到 /mnt目录，可以敲以下命令：

```
# guestmount -a centos63_desktop.qcow2 -m /dev/vg_centosbase/lv_root --rw /mnt
```

如果我们预先不知道镜像内的分区挂载点，可以使用-i（检阅）参数告诉 guestmount 自动检测挂载点：

```
# guestmount -a centos63_desktop.qcow2 -i --rw /mnt
```

当挂载完成后，我们可以干一些诸如使用rpm命令查看软件包安装列表等类似的事：

```
# rpm -qa --dbpath /mnt/var/lib/rpm
```

当一切都完成后，卸载它：

```
# umount /mnt
```

## virt-\* tools

The libguestfs project has a number of other useful tools, including:

- [virt-edit](#) for editing a file inside of an image.
- [virt-df](#) for displaying free space inside of an image.
- [virt-resize](#) for resizing an image.
- [virt-sysprep](#) for preparing an image for distribution (for example, delete SSH host keys, remove MAC address info, or remove user accounts).
- [virt-sparsify](#) for making an image sparse.
- [virt-p2v](#) for converting a physical machine to an image that runs on KVM.
- [virt-v2v](#) for converting Xen and VMWare images to KVM images.

## 修改镜像内单个文件

这个示例演示了如何使用virt-edit修改文件，这个命令既可以通过使用-a标记后跟文件名作为参数，也可以通过使用-d后跟虚拟机的domain名称作为参数。下面的示例演示如何修改在libvirt中运行的domain名称为instance-000000e1的虚拟机内/etc/shadow文件：

```
# virsh shutdown instance-000000e1
# virt-edit -d instance-000000e1 /etc/shadow
# virsh start instance-000000e1
```

## 改变镜像大小

Here is an example of how to use virt-resize to resize an image. Assume we have a 16GB Windows image in qcow2 format that we want to resize to 50GB. First, we use virt-filesystems to identify the partitions:  
# virt-filesystems --long --parts --blkdevs -h -a /  
data/images/win2012.qcow2 Name Type MBR Size Parent /dev/sda1 partition 07 350M /dev/  
sda /dev/sda2 partition 07 16G /dev/sda /dev/sda device - 16G -

在这个示例中，/dev/sda2是我们需要修改大小的分区，我们创建一个新的qcow2镜像并且使用virt-resize命令写入一个修改大小后的原始镜像文件到新镜像文件内：

```
# qemu-img create -f qcow2 /data/images/win2012-50gb.qcow2 50G
# virt-resize --expand /dev/sda2 /data/images/win2012.qcow2
  /data/images/win2012-50gb.qcow2
Examining /data/images/win2012.qcow2 ...
*****
```

```
Summary of changes:  
  
/dev/sda1: This partition will be left alone.  
  
/dev/sda2: This partition will be resized from 15.7G to 49.7G. The  
filesystem ntfs on /dev/sda2 will be expanded using the  
'ntfsresize' method.  
  
*****  
Setting up initial partition table on /data/images/win2012-50gb.qcow2 ...  
Copying /dev/sda1 ...  
100%  
#  
00:00  
Copying /dev/sda2 ...  
100%  
#  
00:00  
Expanding /dev/sda2 using the 'ntfsresize' method ...  
  
Resize operation completed with no errors. Before deleting the old  
disk, carefully check that the resized disk boots and works correctly.
```

## loop(环回)设备，kpartx,网络块设备

如果你不能使用libguestfs，你可以使用loop(环回)设备，kpartx,网络块设备直接挂载镜像文件系统到宿主机系统。



### 警告

Mounting untrusted guest images using the tools described in this section is a security risk, always use libguestfs tools such as guestfish and guestmount if you have access to them. See [A reminder why you should never mount guest disk images on the host OS](#) by Daniel Berrang é for more details.

## 挂载裸镜像（无LVM）

If you have a raw virtual machine image that is not using LVM to manage its partitions, use the losetup command to find an unused loop device.

```
# losetup -f  
/dev/loop0
```

在这个示例中，/dev/loop0是空闲的，关联裸镜像到环回设备：

```
# losetup /dev/loop0 fedora17.img
```

如果镜像只有一个分区，你可以直接挂载环回设备：

```
# mount /dev/loop0 /mnt
```

如果镜像有多个分区，使用kpartx显示分区为单独的设备（如：/dev/mapper/loop0p1），然后挂载对应根文件系统的分区：

```
# kpartx -av /dev/loop0
```

如果镜像显示有三个分区，对应的每个分区会有新的设备文件创建：

```
$ ls -l /dev/mapper/loop0p*
brw-rw---- 1 root disk 43, 49 2012-03-05 15:32 /dev/mapper/loop0p1
brw-rw---- 1 root disk 43, 50 2012-03-05 15:32 /dev/mapper/loop0p2
brw-rw---- 1 root disk 43, 51 2012-03-05 15:32 /dev/mapper/loop0p3
```

要挂载第二个分区，用 root 执行：

```
# mkdir /mnt/image
# mount /dev/mapper/loop0p2 /mnt
```

当所有操作完成后，清理操作：

```
# umount /mnt
# kpartx -d /dev/loop0
# losetup -d /dev/loop0
```

## 挂载裸镜像（有LVM）

If your partitions are managed with LVM, use losetup and kpartx as in the previous example to expose the partitions to the host.

```
# losetup -f
/dev/loop0
# losetup /dev/loop0 rhel62.img
# kpartx -av /dev/loop0
```

下一步，你需要用 vgscan 命令识别 LVM 卷组然后使用 vgchange 激活卷：

```
# vgscan
Reading all physical volumes. This may take a while...
  Found volume group "vg_rhel62x8664" using metadata type lvm2
# vgchange -ay
  2 logical volume(s) in volume group "vg_rhel62x8664" now active
# mount /dev/vg_rhel62x8664/lv_root /mnt
```

当完成操作后，清理命令“

```
# umount /mnt
# vgchange -an vg_rhel62x8664
# kpartx -d /dev/loop0
# losetup -d /dev/loop0
```

## 挂载 qcow2 镜像（无LVM）

你需要加载 nbd（网络块设备）内核模块来挂载 qcow2 镜像文件。下面将加载满足我们需求的支持16个块设备，以root用户执行：

```
# modprobe nbd max_part=16
```

假设第一个块设备(/dev/nbd0)当前未被占用，我们使用qemu-nbd 和 partprobe 列出磁盘分区。以root用户执行：

```
# qemu-nbd -c /dev/nbd0 image.qcow2
# partprobe /dev/nbd0
```

如果镜像有三个分区（/boot, /, swap），对应的每个分区应该有新设备文件建立：

```
$ ls -l /dev/nbd3*
brw-rw---- 1 root disk 43, 48 2012-03-05 15:32 /dev/nbd0
brw-rw---- 1 root disk 43, 49 2012-03-05 15:32 /dev/nbd0p1
brw-rw---- 1 root disk 43, 50 2012-03-05 15:32 /dev/nbd0p2
brw-rw---- 1 root disk 43, 51 2012-03-05 15:32 /dev/nbd0p3
```



## 注意

如果你选择的网络块设备已经被占用，最开始的 qemu-nbd 命令将会失败，并且 /dev/nbd3p{1,2,3} 设备文件也不会被建立。

如果镜像分区没有使用LVM，那么可以直接挂载他们：

```
# mkdir /mnt/image
# mount /dev/nbd3p2 /mnt
```

一切操作完成后，清理命令：

```
# umount /mnt
# qemu-nbd -d /dev/nbd0
```

## 挂载 qcow2 镜像（有LVM）

如果镜像分区使用LVM管理，在你使用 qemu-nbd 和 partprobe 命令后，你需要使用 vgscan 和 vgchange -ay 来列出LVM分区设备以便这些分区能被挂载：

```
# modprobe nbd max_part=16
# qemu-nbd -c /dev/nbd0 image.qcow2
# partprobe /dev/nbd0# vgscan
Reading all physical volumes. This may take a while...
Found volume group "vg_rhel62x8664" using metadata type lvm2
# vgchange -ay
2 logical volume(s) in volume group "vg_rhel62x8664" now active
# mount /dev/vg_rhel62x8664/lv_root /mnt
```

完成后，清理命令：

```
# umount /mnt
# vgchange -an vg_rhel62x8664
# qemu-nbd -d /dev/nbd0
```

# 第 5 章 手动创建镜像

检查 libvirt 默认网络是否运行 .....	20
使用 virt-manager X11 图形界面 .....	20
使用 virt-install 且使用本地 VNC 客户端连接 .....	21
示例：CentOS 镜像 .....	22
示例：Ubuntu 镜像 .....	29
示例：Fedora 镜像 .....	35
示例：微软Windows镜像 .....	38
示例：FreeBSD 镜像 .....	39

创建新的虚拟机镜像是安装完 OpenStack 意外的事情。你在你的系统上手工创建新虚拟机镜像并且上传到云。

To create a new image, you will need the installation CD or DVD ISO file for the guest operating system. You'll also need access to a virtualization tool. You can use KVM for this. Or, if you have a GUI desktop virtualization tool (such as, VMware Fusion or VirtualBox), you can use that instead and just convert the file to raw once you are done.

When you create a new virtual machine image, you will need to connect to the graphical console of the hypervisor, which acts as the virtual machine's display and allows you to interact with the guest operating system's installer using your keyboard and mouse. KVM can expose the graphical console using the [VNC](#) (Virtual Network Computing) protocol or the newer [SPICE](#) protocol. We'll use the VNC protocol here, since you're more likely to be able to find a VNC client that works on your local desktop.

## 检查 libvirt 默认网络是否运行

在使用 libvirt 启动虚拟机前，检查它的 "default" 默认网络是否启动。虚拟机要连接到外网，它的默认网络必须激活。启动 libvirt 默认网络将创建 linux 网桥(通常名称是virbr0)，iptables 规则，以及dhcp服务器进程 dnsmasq 。

检查 "default" 默认网络是否激活，使用 virsh net-list 命令并查看 "default" 网络是否是启用状态。

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

如果网络未激活，输入以下命令：

```
# virsh net-start default
```

## 使用 virt-manager X11 图形界面

If you plan to create a virtual machine image on a machine that can run X11 applications, the simplest way to do so is to use the virt-manager GUI, which is installable as the virt-manager package on both Fedora-based and Debian-based systems. This GUI has an embedded VNC client that will let you view and interact with the guest's graphical console.

如果你在无图形的服务器上创建虚拟机镜像，但在你本地机器有 X server 启动，你可以用 virt-manager，通过使用 ssh X11 转发访问图形界面。因为 virt-manager 和 libvirt 直接交互，你通常需要 root 权限来访问。如果你能直接以 root 用户（或者有权限和 libvirt 交互的用户） ssh 访问服务器，使用以下命令：

```
$ ssh -X root@server virt-manager
```

如果你的账户没有权限和 libvirt 交互，但是有 sudo 权限，运行：

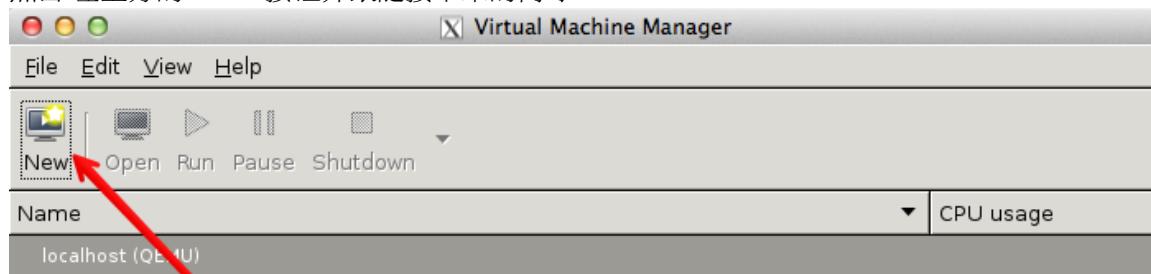
```
$ ssh -X root@server  
$ sudo virt-manager
```



### 注意

-X 参数传递给 ssh 将启用 ssh X11 转发。如果不能工作，尝试用 -Y 参数。

点击左上方的“New”按钮并跟随接下来的向导。



，你将看到一系列允许你指定虚拟机的详细信息的对话框。



### 注意

使用 qcow2 格式镜像你应该勾选 ‘customize before install’，然后跳转到磁盘属性并且确保明确选择了 qcow2 格式。这将保证虚拟机磁盘大小正确。

## 使用 virt-install 且使用本地 VNC 客户端连接

如果你不想使用 virt-manager（例如，你不想安装依赖包到你的服务器，你本地机器没有 X server 或者 X11 转发工作不正常），你可以使用 virt-install 工具通过 Libvirt 启动虚拟机，然后使用本地 VNC 客户端连接到虚拟机的图形控制台。

Because VNC is a standard protocol, there are multiple clients available that implement the VNC spec, including [TigerVNC](#) (multiple platforms), [TightVNC](#) (multiple platforms), [RealVNC](#) (multiple platforms), [Chicken](#) (Mac OS X), [Krdde](#) (KDE), and [Vinagre](#) (GNOME).

The following example shows how to use the qemu-img command to create an empty image file, and virt-install command to start up a virtual machine using that image file. As root:

```
# qemu-img create -f qcow2 /data/centos-6.4.qcow2 10G
```

```
# virt-install --virt-type kvm --name centos-6.4 --ram 1024  
--cdrom=/data/CentOS-6.4-x86_64-netinstall.iso  
--disk path=/data/centos-6.4.qcow2,size=10,format=qcow2  
--network network=default  
--graphics vnc,listen=0.0.0.0 --noautoconsole  
--os-type=linux --os-variant=rhel6  
  
Starting install...  
Creating domain...  
00:00 | 0 B  
Domain installation still in progress. You can reconnect to  
the console to complete the installation process.
```

KVM 虚拟化使用 centos-6.4 名称，1024MB 内存启动虚拟机。虚拟机还有一个关联文件 /data/CentOS-6.4-x86\_64-netinstall.iso 的虚拟的 CD-ROM，并拥有 10GB 的 qcow2 格式的硬盘，硬盘文件位置在 /data/centos-6.4.qcow2。虚拟机配置了网络使用 libvirt 的默认网络。且 VNC 服务监听所有的网卡，并且 libvirt 不会自动启动 VNC 客户端也不会显示字符界面控制台（--no-autoconsole）。最后，libvirt 将尝试以 RHEL 6.x 发行版来优化虚拟机配置。

运行 `virt-install --os-variant list` 命令查看 `--os-variant` 允许的选项范围。

使用 `virsh vncdisplay vm-name` 命令获取 VNC 端口号。

```
# virsh vncdisplay centos-6.4  
:1
```

在上面的示例中，虚拟机 centos-6.4 使用 VNC 显示器 :1，对应的 TCP 端口是 5901。你应该使用本地 VNC 客户端连接到远程服务器的 :1 显示器并且完成安装步骤。

## 示例：CentOS 镜像

这个示例演示如何安装 CentOS 镜像，主要是 CentOS 6.4。由于不同版本的 CentOS 安装过程稍有差别，因此其他版本安装过程可能和示例中有不一样的地方。

### 下载 CentOS 安装 ISO

1. Navigate to the [CentOS mirrors](#) page.
2. 单击镜像中靠近右变的一列的其中一个 HTTP 链接。
3. 单击要使用的 CentOS 版本的文件夹链接。例如， 6.4 / 。
4. 点击isos/ 文件夹的链接。
5. 要下载64位的文件点击 x86\_64/ 文件夹链接。
6. 点击你想下载的网络安装ISO镜像文件，例如：CentOS-6.4-x86\_64-netinstall.iso 就是一个好的选择，因为它文件尺寸更小，在安装的时候，其余的包可以通过网络下载。

## 启动安装程序

通过使用前面的章节介绍的 `virt-manager` 或者 `virt-install` 命令都可以启动安装程序，如果你使用 `virt-install` 记得连接你的 VNC 客户端到虚拟机。

假设：

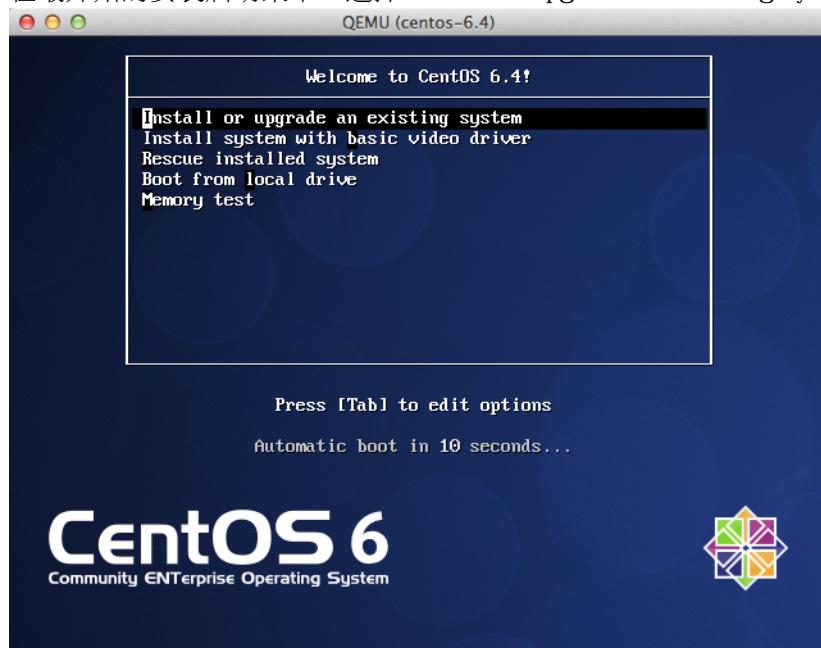
- 你的虚拟机名称是 centos-6.4；在你使用 virsh 命令操作虚拟机状态的时候你需要使用这个名称。
- 你保存下载的网络安装ISO镜像在 /data/isos 目录。

如果你使用 virt-install，命令应该如下所示：

```
# qemu-img create -f qcow2 /tmp/centos-6.4.qcow2 10G
# virt-install --virt-type kvm --name centos-6.4 --ram 1024
--disk /tmp/centos-6.4.qcow2,format=qcow2
--network network=default
--graphics vnc,listen=0.0.0.0 --noautoconsole
--os-type=linux --os-variant=rhel6
--extra-args="console=tty0 console=ttyS0,115200n8 serial"
--location=/data/isos/CentOS-6.4-x86_64-netinstall.iso
```

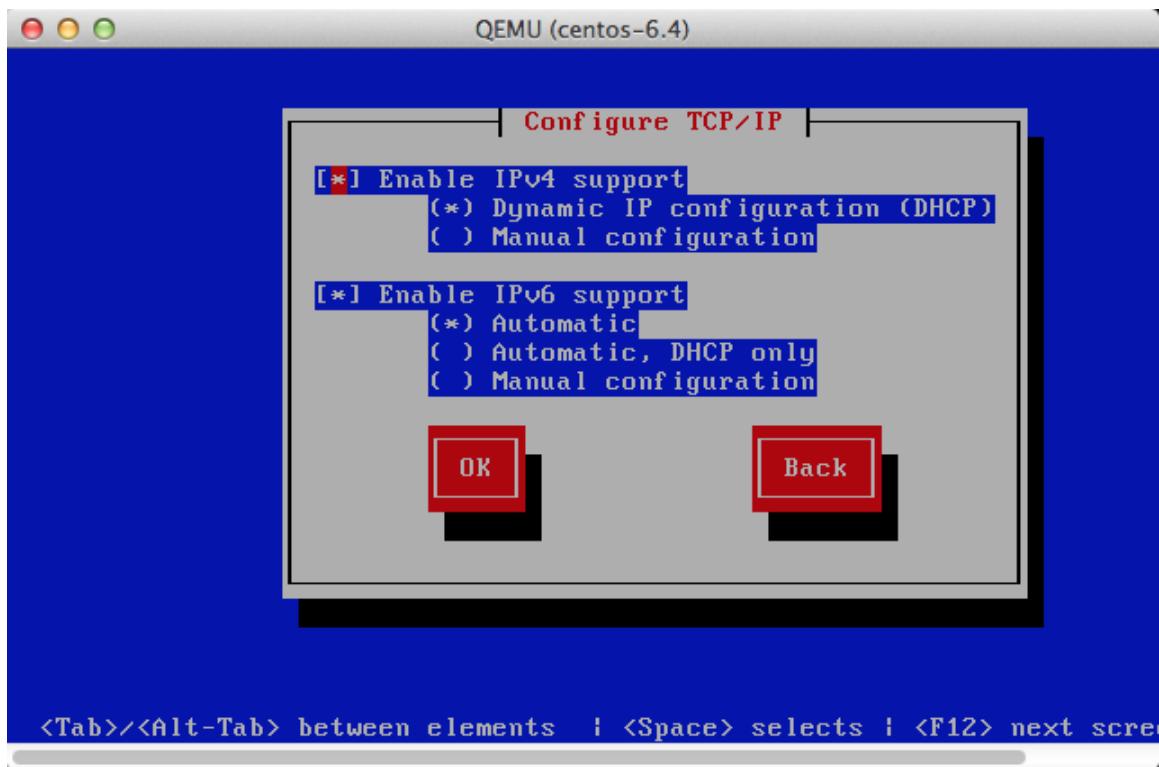
## 安装步骤

在最开始的安装启动菜单，选择 Install or upgrade an existing system，这事默认的。

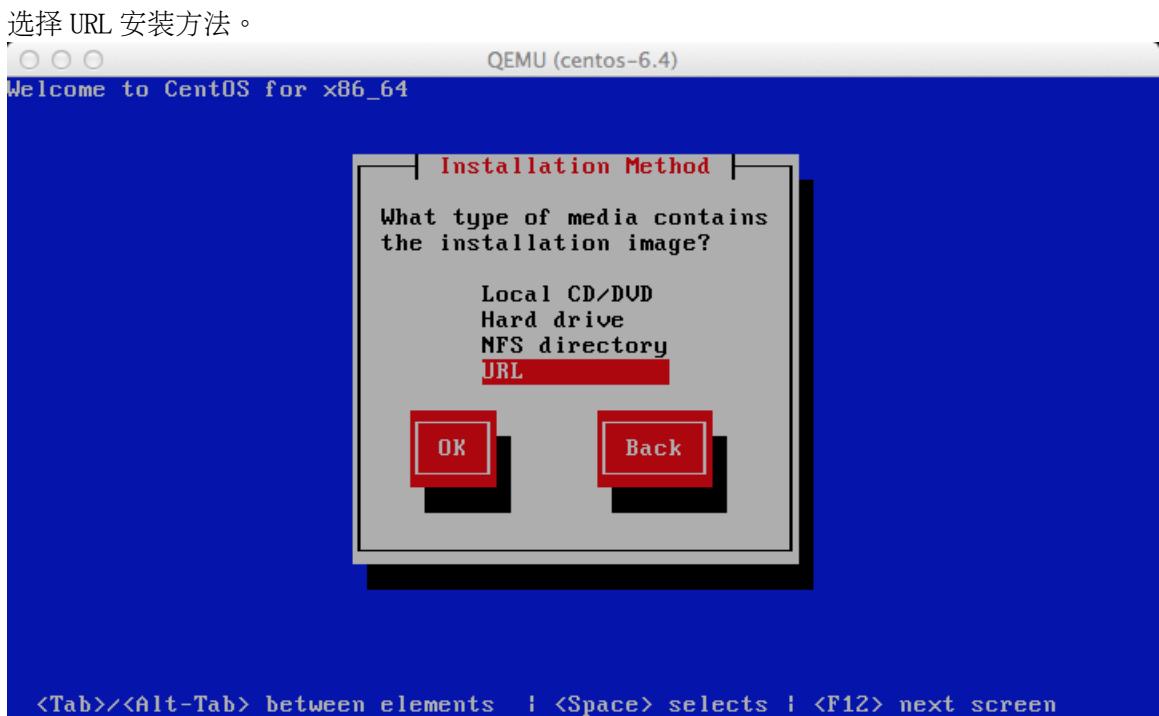


## 配置 TCP/IP

默认的 TCP/IP 就OK了，尤其是，确认 IPv4 支持项启用了 DHCP，默认是启用的。



将安装程序指向 CentOS 网站。



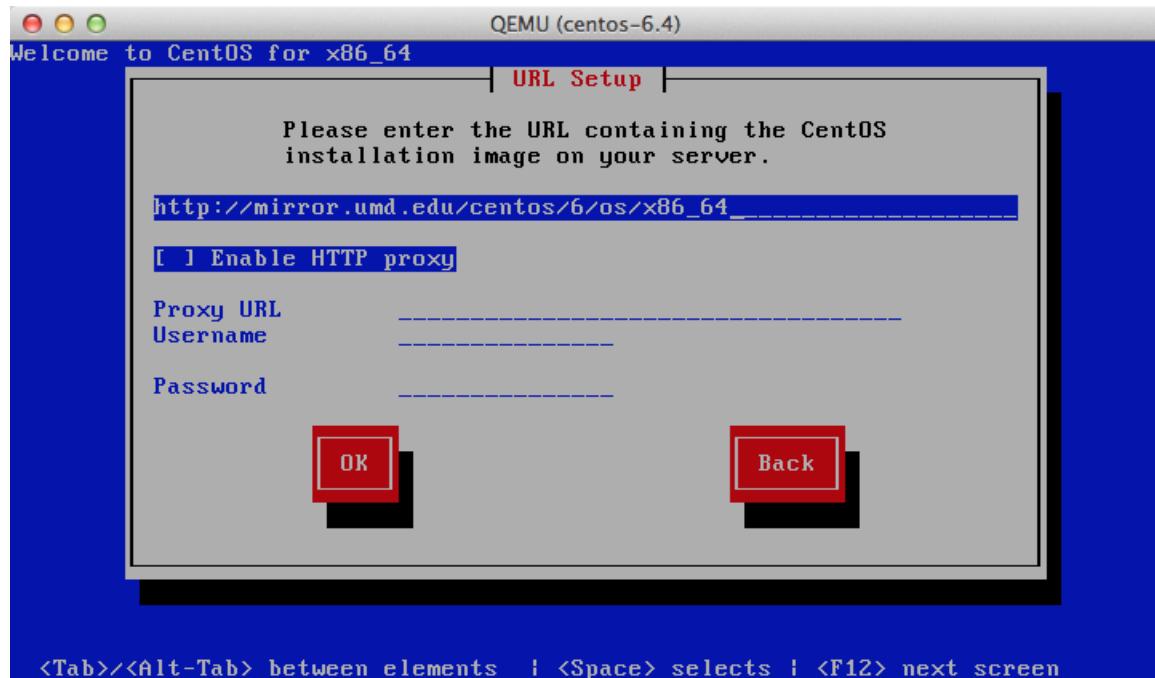
Depending on the version of CentOS, the net installer requires the user to specify either a URL or the web site and a CentOS directory that corresponds to one of the CentOS mirrors.

If the installer asks for a single URL, a valid URL might be `http://mirror.umd.edu/centos/6/os/x86_64`.



## 注意

考虑使用其他镜像作为 `mirror.umd.edu` 的替代，国内推荐使用 `mirrors.163.com`。



如果安装程序要求分别指定网站地址和 CentOS 目录，你可以这样输入：

- 网站地址：`mirror.umd.edu`
- CentOS 目录：`centos/6/os/x86_64`

See [CentOS mirror page](#) to get a full list of mirrors, click on the "HTTP" link of a mirror to retrieve the web site name of a mirror.

## 存储设备

如果提示你使用哪种类型存储设备，选择 Basic Storage Devices。

## 主机名

安装程序让你输入主机名称，默的认(`localhost.localdomain`) 也可以，随后你将安装 cloud-init 包，它将在镜像将来启动时设置主机名。

## 磁盘分区

磁盘分区有好几个选项，默认使用LVM分区，会创建(`/boot`, `/`, `swap`)三个分区，默认设置就可以正常工作。另外你也可以只建立一个 ext4 分区挂载到 `"/"`，这也可以正常工作。

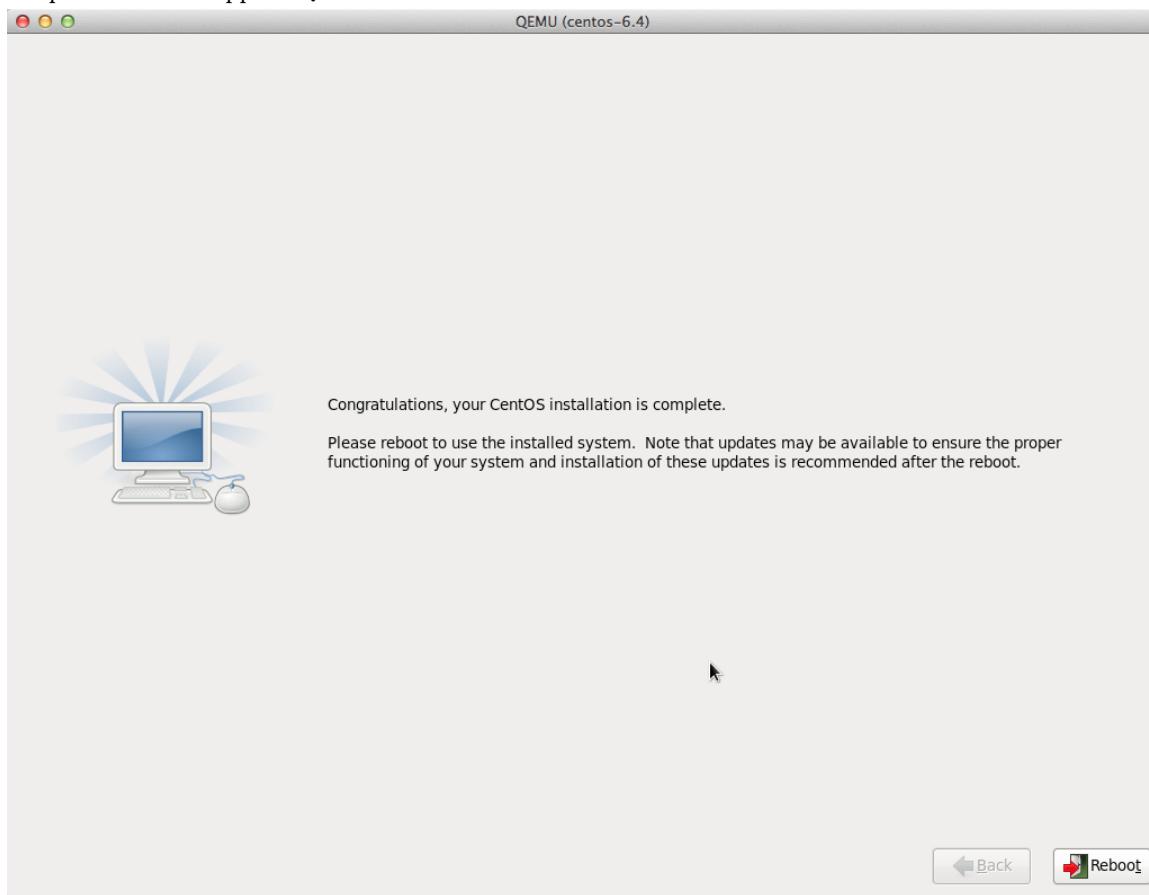
如果你不太确定，使用默认的分区方案就可以了，因为没有一定说某个方案就比其他方案要优秀。

## 安装步骤

使用默认设置，完成安装程序设置步骤，最后最简单的事选择“Basic Server”软件包组（以前版本的 CentOS 可能叫做“Server”），它将默认安装 SSH 服务。

## 卸载 CD-ROM 并重启

When the installation has completed, the Congratulations, your CentOS installation is complete screen appears.



使用 virsh 卸载磁盘，libvirt 需要挂接一个空磁盘到之前挂接 CDROM 的地方，设备名应该是 hdc。可以使用 `dom dumpxml vm-image` 命令来确认。

```
# virsh dumpxml centos-6.4
<domain type='kvm'>
  <name>centos-6.4</name>
...
  <disk type='block' device='cdrom'>
    <driver name='qemu' type='raw'/>
    <target dev='hdc' bus='ide' />
    <readonly/>
    <address type='drive' controller='0' bus='1' target='0' unit='0' />
  </disk>
```

```
</domain>
```

以root用户使用virsh命令，执行下面的步骤完成挂载空磁盘并重启。如果你使用virt-manager安装，下面的命令也起相同作用，你也可以使用图形界面卸载磁盘并手工停止然后启动虚拟机。

```
# virsh attach-disk --type cdrom --mode readonly centos-6.4 "" hdc
# virsh destroy centos-6.4
# virsh start centos-6.4
```

## 登陆到新创建的镜像内

安装后的第一次启动，会提示认证工具（我咋没遇到过啊，文档bug吧）。选择Exit。然后，以root用户登录。

## 安装 ACPI 服务

为了让虚拟化层能重启和关闭虚拟机，必须在虚拟机内安装并运行 acpid 服务。

在 CentOS 虚拟机内运行以下命令安装 ACPI 服务并配置为系统启动时自动启动。

```
# yum install acpid
# chkconfig acpid on
```

## 配置获取metadata

虚拟机在启动的时候需要和元数据服务交互以便完成几个任务，例如，虚拟机必须获取ssh公钥以及运行用户数据脚本。确保虚拟机能完成这些任务，使用以下几个方法之一：

- Install a cloud-init RPM, which is a port of the Ubuntu [cloud-init](#) package. This is the recommended approach.
- 修改 /etc/rc.local 文件从元数据服务获取信息，具体将在下一届描述。

## 使用 cloud-init 获取公钥

cloud-init 包自动从元数据服务获取公钥并保存到账户，你可以在 CentOS 虚拟机添加 EPEL 安装源来安装 cloud-init 包。

```
# yum install http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
# yum install cloud-init
```

不同的发行版保存的账户不同，在基于 Ubuntu 的虚拟机，账户是 ubuntu，基于 Fedora 的虚拟机，账户是 ec2-user。

你可以编辑 /etc/cloud/cloud.cfg 文件添加一行修改 cloud-init 使用的账户名，例如，添加以下这行到配置文件，来配置 cloud-init 存放公钥到 admin 账户。

```
user: admin
```

## 编写脚本获取公钥（如果未使用cloud-init）

如果你不能在镜像中安装cloud-init软件包，就获取ssh公钥并添加到root帐号中，编辑/etc/rc.d/rc.local文件，在touch /var/lock/subsys/local之前添加以下内容：

```
if [ ! -d /root/.ssh ]; then
    mkdir -p /root/.ssh
```

```

chmod 700 /root/.ssh
fi

# Fetch public key using HTTP
ATTEMPTS=30
FAILED=0
while [ ! -f /root/.ssh/authorized_keys ]; do
curl -f http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
>/tmp/metadata-key 2>/dev/null
if [ $? -eq 0 ]; then
cat /tmp/metadata-key >> /root/.ssh/authorized_keys
chmod 0600 /root/.ssh/authorized_keys
restorecon /root/.ssh/authorized_keys
rm -f /tmp/metadata-key
echo "Successfully retrieved public key from instance metadata"
echo "*****"
echo "AUTHORIZED KEYS"
echo "*****"
cat /root/.ssh/authorized_keys
echo "*****"
done

```



## 注意

有些 VNC 客户会使用分号 (;) 替换冒号(:) 以及用中划线 (-) 替换下划线(\_)，确认你输入的是 http: 而不是 http; ， authorized\_keys 而不是 authorized-keys。



## 注意

前面的脚本仅仅只是从元数据服务获取了 ssh 的公钥，并未获取用户数据，用户数据是用户申请新虚拟机可以传递到虚拟机内的可选数据，通常用于在虚拟机启动时做一些定制化工作。

As the OpenStack metadata service is compatible with version 2009-04-04 of the Amazon EC2 metadata service, consult the Amazon EC2 documentation on [Using Instance Metadata](#) for details on how to get user data.

## 禁用 zeroconf 路由

要让虚拟机访问元数据服务，必须禁用 zeroconf 路由：

```
# echo "NOZEROCONF=yes" >> /etc/sysconfig/network
```

## 配置 console

要使 nova console-log 命令在 CentOS 6.x 生效，你需要添加以下文字到 /boot/grub/menu.lst 文件：

```

serial --unit=0 --speed=115200
terminal --timeout=10 console serial
# Edit the kernel line to add the console entries
kernel ... console=tty0 console=ttyS0,115200n8

```

## 关闭虚拟机

在虚拟机内，以root用户运行：

```
# /sbin/shutdown -h now
```

## 清理（删除 MAC 地址相关信息）

操作系统会在/etc/sysconfig/network-scripts/ifcfg-eth0 和 /etc/udev/rules.d/70-persistent-net.rules 这类文件记录下网卡MAC地址，但是，虚拟机的网卡MAC地址在每次虚拟机创建的时候都会不同，因此这些信息必须从配置文件删除掉。

目前有 virt-sysprep 工具可以完成清理虚拟机镜像内的 MAC 地址相关的信息。

```
# virt-sysprep -d centos-6.4
```

## 删除 libvirt 虚拟机域

Now that you can upload the image to the Image service, you no longer need to have this virtual machine image managed by libvirt. Use the virsh undefine vm-image command to inform libvirt:

```
# virsh undefine centos-6.4
```

## 镜像准备完成

The underlying image file that you created with qemu-img create is ready to be uploaded. For example, you can upload the /tmp/centos-6.4.qcow2 image to the Image service.

## 示例：Ubuntu 镜像

这个示例安装Ubuntu 14.04 (Trusty Tahr) 版本镜像。创建其他版本的Ubuntu镜像，注意下面安装过程的标记不同的地方。

## 下载一个 Ubuntu 安装 ISO

Because the goal is to make the smallest possible base image, this example uses the network installation ISO. The Ubuntu 64-bit 14.04 network installer ISO is at <http://archive.ubuntu.com/ubuntu/dists/trusty/main/installer-amd64/current/images/netboot/miniso>.

## 开始安装

使用在之前示例中讲解的 virt-manager 或者 virt-install 命令。如果你使用 virt-install ，记得使用 VNC 客户端连接到虚拟机。

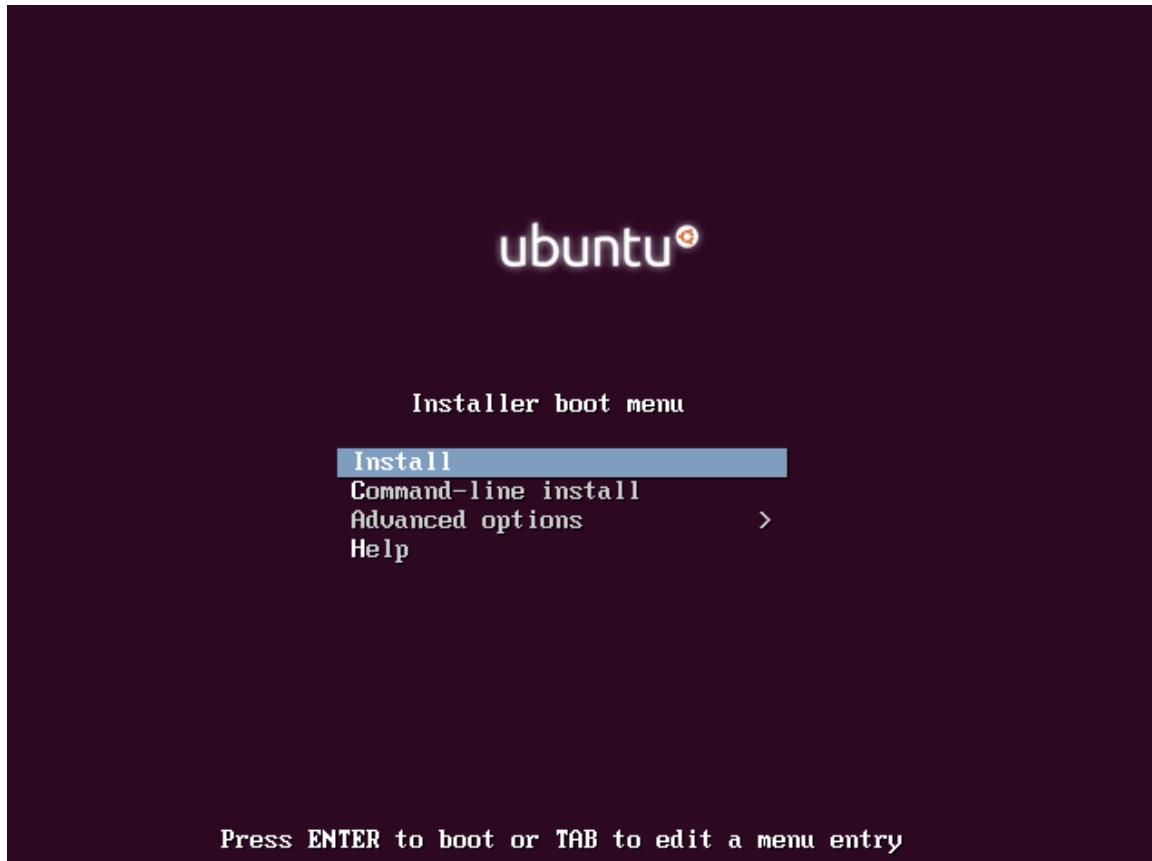
假设你的虚拟机名称是 trusty ，当你使用 virsh 命令操作虚拟机状态的时候你需要虚拟机名称。

如果你使用 virt-manager 命令，示例命令如下：

```
# qemu-img create -f qcow2 /tmp/trusty.qcow2 10G
# virt-install --virt-type kvm --name trusty --ram 1024
--cdrom=/data/isos/trusty-64-miniso
--disk /tmp/trusty.qcow2,format=qcow2
--network network=default
--graphics vnc,listen=0.0.0.0 --noautoconsole
--os-type=linux --os-variant=ubuntutrusty
```

## 安装步骤

在安装初始启动菜单中，选择 Install 。安装过程提示中，默认的即可。



## 主机名

安装程序要求填入主机名。默认的 (ubuntu) 即可。后面我们将安装 cloud-init 包，cloud-init 包将在将来使用镜像创建新虚拟机启动的时候设置主机名。

## 选择镜像站点

选择默认的镜像站点即可。

## 安装步骤

安装过程中，使用默认的选项即可，当提示输入用户名时，使用默认的 (ubuntu) 即可。

## 磁盘分区

目前有几个分区选项。默认的安装程序使用 LVM 分区，将创建三个分区 (/boot, /, swap)，使用默认的即可。作为可选的，你也可以创建单个 ext4 分区，挂载 "/"。

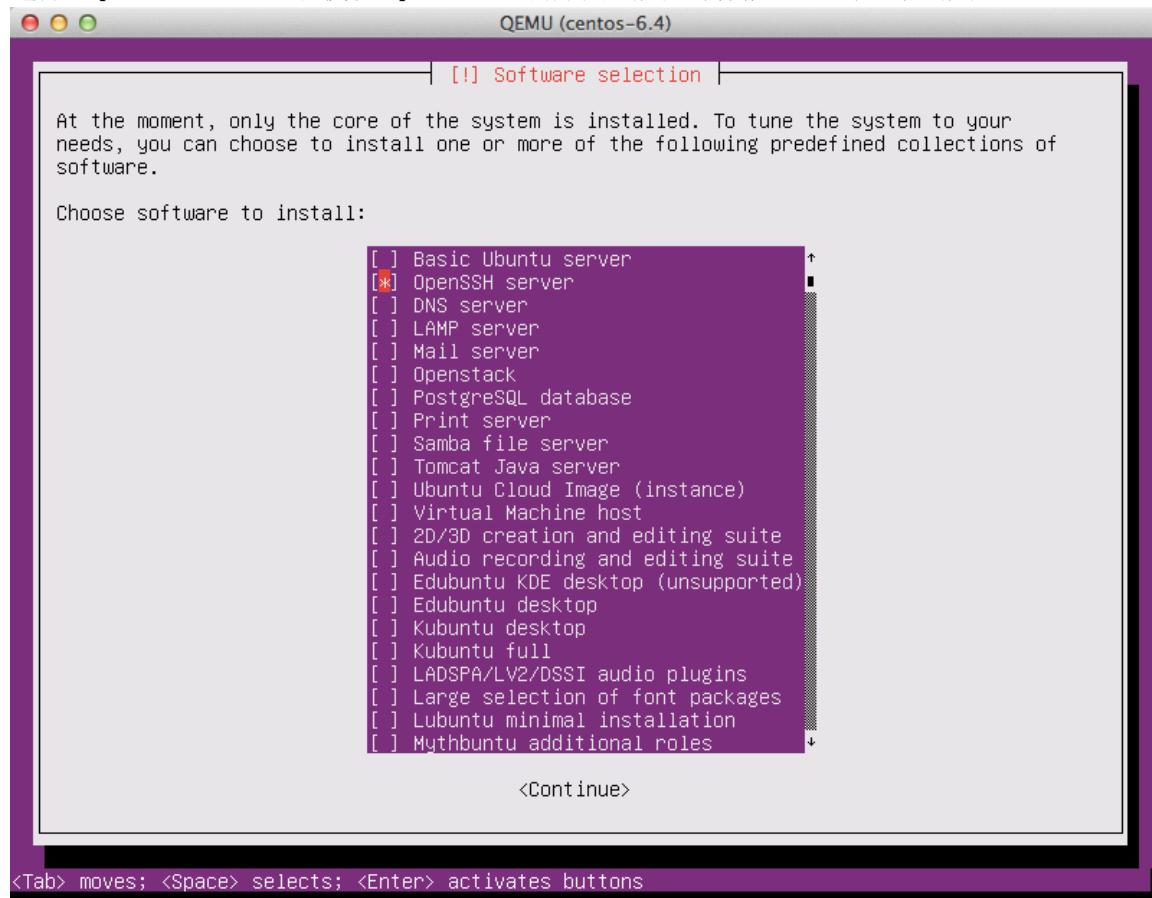
如果你不太确定如何分区，我们推荐你使用安装程序默认的分区方案，因为目前也没有任何方案完全优于其他分区方案。

## 自动更新

Ubuntu 安装程序询问你如何设置系统更新。这个选项依赖于你的使用环境，如果将来创建的虚拟机能链接到因特网，我们推荐 “Install security updates automatically”。

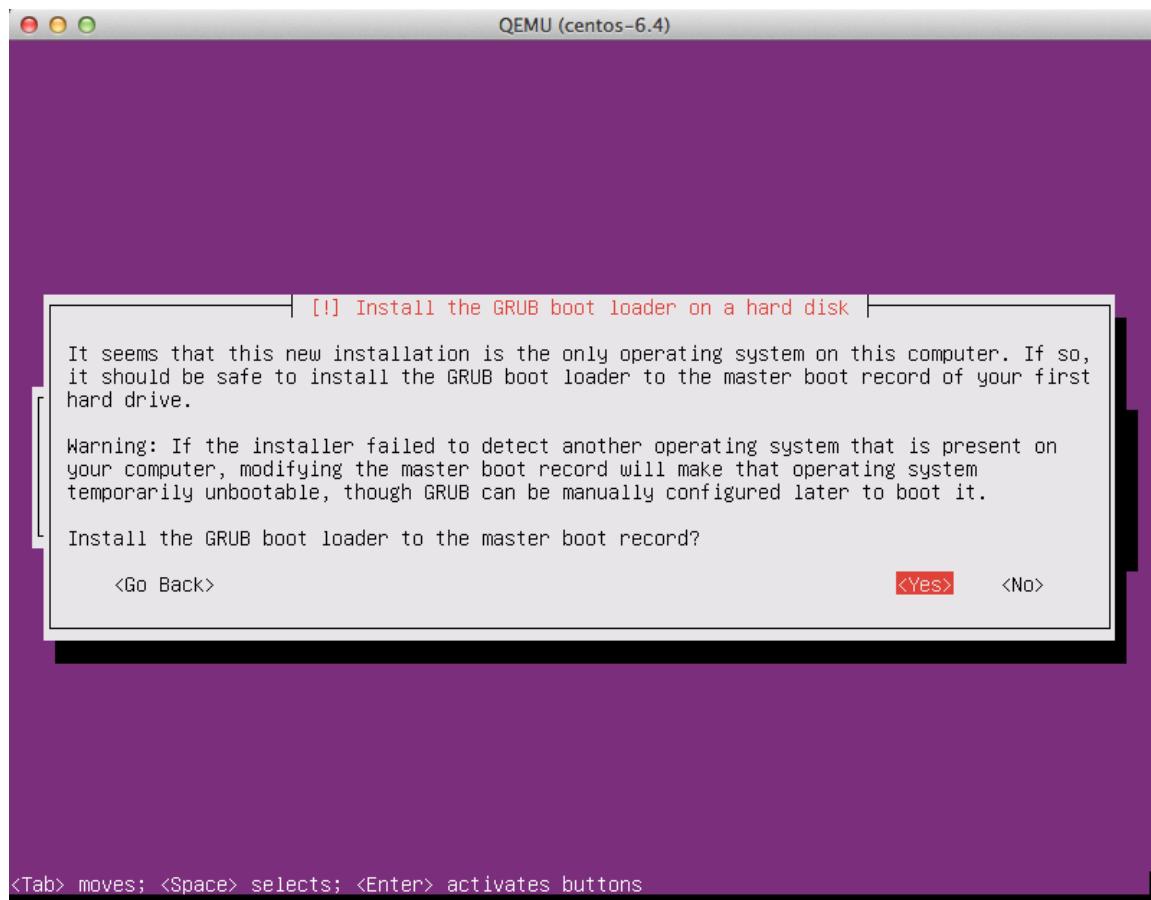
## 软件选择：OpenSSH 服务

选择 “OpenSSH server” 以便在 OpenStack 中启动虚拟机时你能 SSH 登陆到虚拟机。



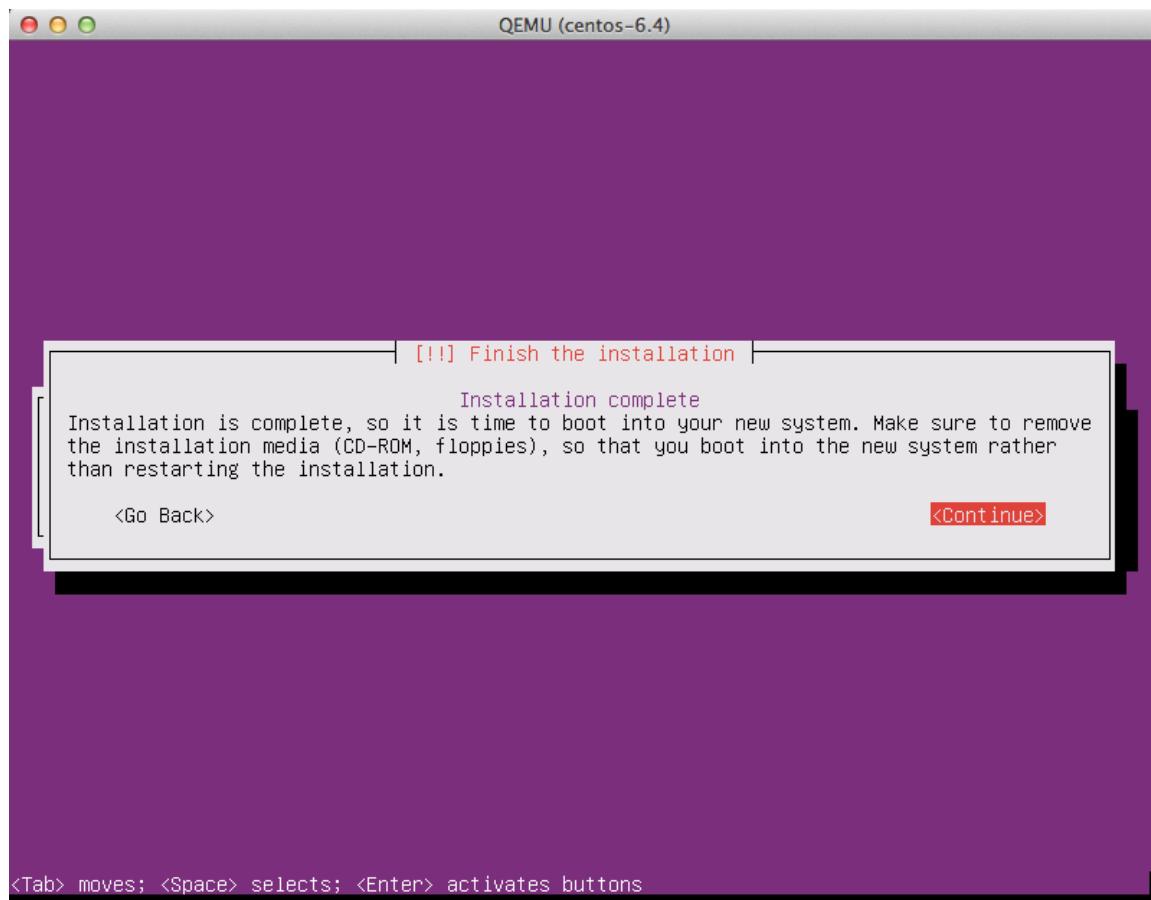
## 安装 GRUB 引导程序

当提示安装 GRUB 引导程序到硬盘主启动记录时，选择 “Yes”。



## 卸载 CD-ROM 并重启

剩下的步骤中，选择默认的值即可。当安装完成时，你会看到提示移除 CD-ROM。



## 注意

There is a known bug in Ubuntu 14.04; when you select "Continue", the virtual machine will shut down, even though it says it will reboot.

使用 virsh 弹出光驱时，libvirt 要求你挂一个空磁盘在光驱相同的位置。在之前盘符是 hdc。  
你可以使用 dom dumpxml vm-image 命令来完成。

```
# virsh dumpxml trusty
<domain type='kvm'>
  <name>trusty</name>
  ...
  <disk type='block' device='cdrom'>
    <driver name='qemu' type='raw' />
    <target dev='hdc' bus='ide' />
    <readonly />
    <address type='drive' controller='0' bus='1' target='0' unit='0' />
  </disk>
...
</domain>
```

在宿主机上以 root 用户运行以下命令暂停虚拟机，弹出磁盘然后恢复虚拟机。如果你使用 virt-manager ，你需要在图形界面完成这个过程。

```
# virsh start trusty --paused
# virsh attach-disk --type cdrom --mode readonly trusty "" hdc
```

```
# virsh resume trusty
```



## 注意

在前面的示例中，你暂停虚拟机，弹出磁盘，然后恢复虚拟机，理论上，你在 Installation complete 屏幕显示的时候就能弹出磁盘，然后我们测试显示 Ubuntu 安装程序锁定了磁盘，因此，不能弹出磁盘。

## 登陆到新创建的镜像内

当你安装后首次启动时，虚拟机将会询问你认证工具，你只需要选择 'Exit'。然后使用你设置的root密码以root用户登录。

## 安装 cloud-init

cloud-init 脚本将在虚拟机启动的时候搜寻元数据服务获取公钥。公钥将会放在镜像默认用户内。

安装 cloud-init 软件包：

```
# apt-get install cloud-init
```

在创建 Ubuntu 镜像时，cloud-init 必须明确的配置元数据源。OpenStack 元数据服务仿效 Amazon EC2 元数据服务。

运行 dpkg-reconfigure 命令设置镜像 cloud-init 软件包使用的元数据源。当屏幕出现提示时，选择 EC2 数据源。

```
# dpkg-reconfigure cloud-init
```

不同的发行版保存的账户不同，在基于 Ubuntu 的虚拟机，账户是 ubuntu，基于 Fedora 的虚拟机，账户是 ec2-user。

你可以编辑 /etc/cloud/cloud.cfg 文件添加一行修改 cloud-init 使用的账户名，例如，添加以下这行到配置文件，来配置 cloud-init 存放公钥到admin账户。

```
user: admin
```

## 关闭虚拟机

在虚拟机内，以root用户运行：

```
# /sbin/shutdown -h now
```

## 清理（删除 MAC 地址相关信息）

操作系统会在/etc/sysconfig/network-scripts/ifcfg-eth0 和 /etc/udev/rules.d/70-persistent-net.rules 这类文件记录下网卡MAC地址，但是，虚拟机的网卡MAC地址在每次虚拟机创建的时候都会不同，因此这些信息必须从配置文件删除掉。

目前有 virt-sysprep 工具可以完成清理虚拟机镜像内的 MAC 地址相关的信息。

```
# virt-sysprep -d trusty
```

## 删除 libvirt 虚拟机械

Now that the image is ready to be uploaded to the Image service, you no longer need to have this virtual machine image managed by libvirt. Use the virsh undefine vm-image command to inform libvirt:

```
# virsh undefine trusty
```

## 镜像准备完成

The underlying image file that you created with qemu-img create, such as /tmp/trusty.qcow2, is now ready for uploading to the OpenStack Image service.

## 示例：Fedora 镜像

Download a [Fedora](#) ISO image. This procedure lets you create a Fedora 20 image.

1. 使用 virt-install 开始安装，如下：

```
# qemu-img create -f qcow2 fedora-20.qcow2 8G
# virt-install --connect=qemu:///system --network=bridge:virbr0
--extra-args="console=tty0 console=ttyS0,115200 serial rd_NO_PLYMOUTH"
--name=fedora-20 --disk path=/var/lib/libvirt/images/fedora-20.qcow2,format=qcow2,size=10,cache=
none
--ram 2048 --vcpus=2 --check-cpu --accelerate --os-type linux --os-variant fedora19
--hvm --location=http://dl.fedoraproject.org/pub/fedora/linux/releases/20/Fedora/x86_64/os/
--nographics
```

这将启动一个虚拟机并且开始安装过程。

```
Starting install...
Retrieving file .treeinfo... | 2.2 kB 00:00:00 !!!
Retrieving file vmlinuz... | 9.8 MB 00:00:05 !!!
Retrieving file initrd.img... | 66 MB 00:00:37 !!!
Allocating 'fedora-20.qcow2' | 10 GB 00:00:00
Creating domain... | 0 B 00:00:00
Connected to domain fedora-20
Escape character is ^]
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Initializing cgroup subsys cpacct
...
...
[ OK ] Reached target Local File Systems (Pre).
Starting installer, one moment...
anaconda 20.25.15-1 for Fedora 20 started.
=====
```

2. 安装选项选择 VNC 或者 文本模式。

```
Text mode provides a limited set of installation
options. It does not offer custom partitioning for full control over the
disk layout. Would you like to use VNC mode instead?
```

1) Start VNC

2) Use text mode

Please make your choice from above ['q' to quit | 'c' to continue | 'r' to refresh]:

3. 设置时区，网络配置，安装源，root用户密码，以及作为可选项，你还可以创建额外的用户。

4. 设置安装位置：

=====

Probing storage...

Installation Destination

[x] 1) Virtio Block Device: 10.24 GB (vda)

1 disk selected; 10.24 GB capacity; 10.24 GB free ...

Please make your choice from above ['q' to quit | 'c' to continue | 'r' to refresh]: c

=====

Autopartitioning Options

[ ] 1) Replace Existing Linux system(s)

[x] 2) Use All Space

[ ] 3) Use Free Space

Installation requires partitioning of your hard drive. Select what space to use for the install target.

Please make your choice from above ['q' to quit | 'c' to continue | 'r' to refresh]: 2

=====

Autopartitioning Options

[ ] 1) Replace Existing Linux system(s)

[x] 2) Use All Space

[ ] 3) Use Free Space

Installation requires partitioning of your hard drive. Select what space to use for the install target.

Please make your choice from above ['q' to quit | 'c' to continue | 'r' to refresh]: c

=====

Partition Scheme Options

[ ] 1) Standard Partition

[x] 2) LVM

[ ] 3) BTRFS

```
Select a partition scheme configuration.  
Please make your choice from above ['q' to quit | 'c' to continue |  
'r' to refresh]: c  
Generating updated storage configuration  
Checking storage configuration...  
=====
```

5. 以root用户使用virsh命令弹出安装盘并且重启虚拟机。

```
# virsh attach-disk --type cdrom --mode readonly fedora-20 "" hdc  
# virsh destroy fedora-20  
# virsh start fedora-20
```

你也可以使用图形界面手工停止和启动虚拟机来完成以上功能。

6. 安装完成重启后以root用户登录。
7. 在虚拟机上安装并且启用 acpid 服务，以便能够重启或者关闭虚拟机。

在 Fedora 虚拟机内运行一下命令安装并启用 ACPI服务。

```
# yum install acpid  
# chkconfig acpid on
```

8. 在 Fedora 虚拟机添加 EPEL 源以便安装 cloud-init 软件包。

cloud-init 自动从元数据服务器获取公钥并保存到账户内。

```
# yum install http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm  
# yum install cloud-init
```

你可以编辑 /etc/cloud/cloud.cfg 文件添加一行修改 cloud-init 使用的账户名，例如，添加以下这行到配置文件，来配置 cloud-init 存放公钥到admin账户。

```
user: admin
```

9. 禁用默认的 zeroconf 路由。

```
# echo "NOZEROCONF=yes" >> /etc/sysconfig/network
```

10. 要让 nova console-log 命令在 Fedora 20虚拟机上，你需要编辑 /boot/grub/menu.lst 文件添加以下几行。

```
serial --unit=0 --speed=115200  
terminal --timeout=10 console serial  
# Edit the kernel line to add the console entries  
kernel ... console=tty0 console=ttyS0,115200n8
```

11. 在虚拟机内以root用户关闭虚拟机。

```
# /sbin/shutdown -h now
```

12. 清除和删除 MAC 地址。

操作系统会在/etc/sysconfig/network-scripts/ifcfg-eth0 和 /etc/udev/rules.d/70-persistent-net.rules 这类文件记录下网卡MAC地址，但是，虚拟机的网卡MAC地址在每次虚拟机创建的时候都会不同，因此这些信息必须从配置文件删除掉。

使用 virt-sysprep 工具，它将通过几个步骤完成清除镜像内相关记录的MAC地址信息，

```
# virt-sysprep -d fedora-20
```

13. 将虚拟机domain undefine掉，以为你已经完成镜像制作不再需要libvirt进行管理。

```
# virsh undefine fedora-20
```

The underlying image file that you created with qemu-img create is ready to be uploaded to the Image service.

## 示例：微软Windows镜像

这个例子创建一个Windows Server 2012 qcow2镜像，使用virt-install以及KVM虚拟化。

1. 按照这些步骤做安装前准备：

- a. Download a Windows Server 2012 installation ISO. Evaluation images are available on [the Microsoft website](#) (registration required).
- b. Download the signed VirtIO drivers ISO from the [Fedora website](#).
- c. 创建一个15GB大小的磁盘镜像：

```
$ qemu-img create -f qcow2 ws2012.qcow2 15G
```

2. 使用virt-install命令开始 Windows Server 2012 的安装：

```
# virt-install --connect qemu:///system  
--name ws2012 --ram 2048 --vcpus 2  
--network network=default,model=virtio  
--disk path=ws2012.qcow2,format=qcow2,device=disk,bus=virtio  
--cdrom /path/to/en_windows_server_2012_x64_dvd.iso  
--disk path=/path/to/virtio-win-0.1-XX.iso,device=cdrom  
--vnc --os-type windows --os-variant win2k8
```

使用 virt-manager 或 virt-viewer 命令连接到虚拟机开始 Windows 安装。

3. 启用 VirtIO 驱动

Windows 安装程序默认不识别硬盘，当安装程序要求你选择安装位置时，点击加载驱动并浏览文件系统选择 E: WIN8 AMD64 文件夹。Windows 安装程序显示驱动列表，选择 VirtIO SCSI和网络驱动，继续安装。

安装完成，虚拟机重启后，为虚拟机administrator用户设置密码。

4. 以administrator用户登录并开启命令行窗口。

5. 使用以下命令完成VirtIO驱动的安装：

```
C: pnputil -i -a E: WIN8 AMD64 *.INF
```

6. 为了让Cloudbase-Init在系统启动时运行脚本，设置PowerShell执行策略解除限制。

```
C: powershell
```

```
C: Set-ExecutionPolicy Unrestricted
```

## 7. 下载并安装 Cloudbase-Init :

```
C: Invoke-WebRequest -UseBasicParsing http://www.cloudbase.it/downloads/
CloudbaseInitSetup_Stable_x64.msi -OutFile cloudbaseinit.msi
C: .\cloudbaseinit.msi
```

在configuration options窗口，修改以下设置：

- 用户名：Administrator
- 网卡：Red Hat VirtIO Ethernet Adapter
- 日志输出串口：COM1

当安装完成后，在Complete the Cloudbase-Init Setup Wizard 窗口，选择Run Sysprep和Shutdown 复选框，然后点击Finish。

等待虚机关闭。

Your image is ready to upload to the Image service:

```
$ glance image-create --name WS2012 --disk-format qcow2
--container-format bare --is-public true
--file ws2012.qcow2
```

## 示例：FreeBSD 镜像

This example creates a minimal FreeBSD image that is compatible with OpenStack and bsd-cloudinit. The bsd-cloudinit program is independently maintained and in active development. The best source of information on the current state of the project is at <http://pellaeon.github.io/bsd-cloudinit/>.

这个示例中使用 KVM 以及 virtio 驱动做为虚拟化平台，因为这是 Openstack 使用最广泛的平台之一，如果你使用其他平台做为你的云计算虚拟化，在创建中镜像也使用相同的平台。

这个示例显示了如何创建 FreeBSD 10 镜像，要创建 FreeBSD 9.2 镜像，注意以下步骤中提示。

### 创建 FreeBSD 镜像

#### 1. 创建一个虚拟磁盘：

```
$ qemu-img create -f qcow2 freebsd.qcow2 1G
```

FreeBSD 支持的最小磁盘大小是1GB。因为我们的目标是建立一个最小化的基础镜像文件，所以示例使用的最小磁盘大小，足够容纳可选的文档，游戏以及lib32库。要包含 ports 软件，需要添加额外的1GB空间，要包含源码，再添加512M空间。

#### 2. 获取安装ISO文件：

```
$ curl ftp://ftp.freebsd.org/pub/FreeBSD/releases
/amd64/amd64/ISO-IMAGES/10.0/FreeBSD-10.0-RELEASE-amd64-bootonly.iso >
FreeBSD-10.0-RELEASE-amd64-bootonly.iso
```

3. 启动一个虚拟机，使用和你生产环境一样的虚拟化层，虚拟磁盘以及网卡驱动。

下面的命令使用最小内存128MB：

```
$ kvm -smp 1 -m 128 -cdrom FreeBSD-10.0-RELEASE-amd64-bootonly.iso  
-drive if=virtio,file=freebsd.qcow2  
-net nic,model=virtio -net user
```

你可以指定最大1GB内存让安装程序运行更快。

虚拟机必须能访问 Internet 以便下载软件包。



## 注意

通过使用相同的虚拟化层，可以确保仿照你生产环境相同的虚拟设备，然而，如果你有硬件虚拟化代替半虚拟化，可以不需要相同的虚拟化层；你必须使用相同类型的虚拟化硬件因为 FreeBSD 设备名和驱动相关联。如果创建镜像环境中根块设备或者第一个网卡和你生产环境不一样，那么在生产环境中使用镜像会产生错误。

现在你的虚拟机从之前你下载的ISO文件启动并且使用了之前准备好的空白虚拟磁盘。

4. 在虚拟机里，完成以下步骤安装操作系统：

- a. 当出现提示的时候，选择运行ISO于Install模式。
- b. 接受默认键盘分布或者根据需要选择适当的键盘映射。
- c. 提供主机名。如果你使用 bsd-cloudinit，它将在虚拟机启动时使用OpenStack分配给他的主机名覆盖这个值。
- d. 当提示可选doc, games, lib32, ports, and src 系统组件时，选择你需要的组件，当你不选可选的系统组件时也可以完成一个完全功能安装，如前所示，最小化系统的1GB虚拟磁盘支持包含doc, games, and lib32内容。要包含ports 软件组需要至少1GB额外空间，如果你计划安装更多port软件时还需要更多的空间，src（源代码）组需要额外512MB空间。
- e. 配置首选网卡使用DHCP，在这个示例中，使用 virtio 网卡，在虚拟机中命名为vtnet0。
- f. 接受默认的网络镜像。
- g. 设置磁盘分区。

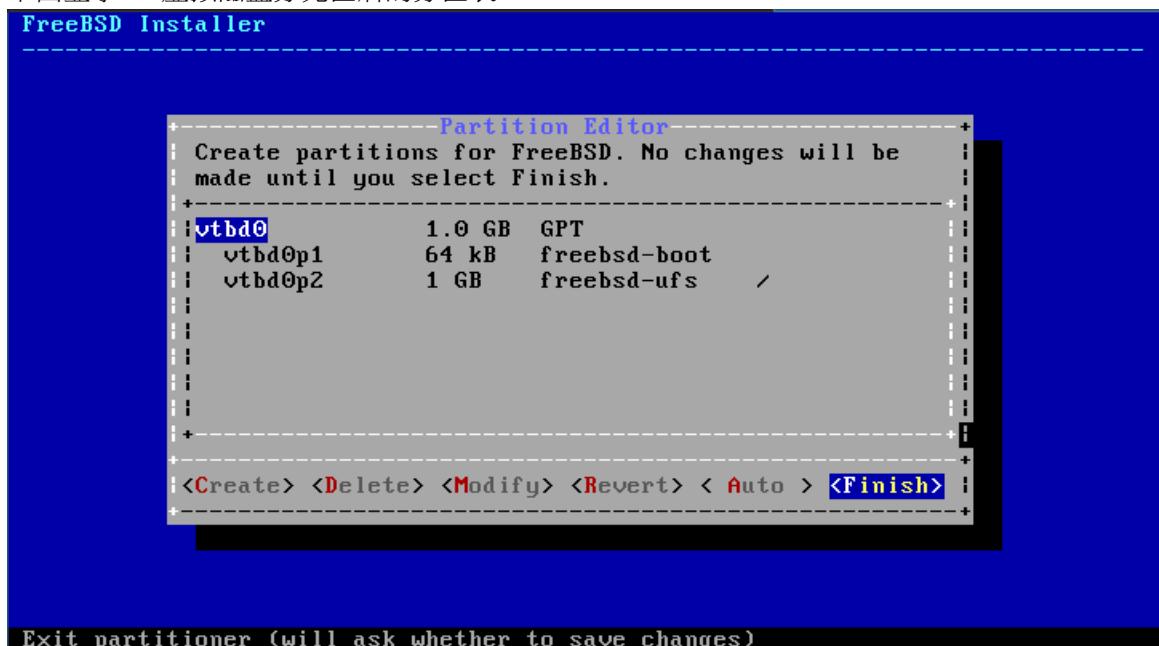
磁盘分区在镜像创建过程中是一个重要的要素，它自动产生默认的分区表会导致 bsd-cloudinit 不起作用。

因为默认的分区会导致 bsd-cloudinit 不能正常工作，你必须手工分区，分区编辑器仅列出块设备，如果你使用 virtio 磁盘驱动，它的名称是 vtbd0。选择这个设备运行 create 三次：

1. 选择 Create 创建分区表，当没有分区表存在时这是默认动作，然后，从列表中选择 GPT GUID Partition Table，这也是默认选择。
2. 创建两个分区：

- 第一个分区：一个64kB freebsd 启动分区，无挂载点。
- 第二个分区： freebsd-ufs 分区挂载点是 / 并使用所有剩下的空余空间。

下图显示1GB虚拟磁盘分完区后的分区表：



选择 Finish 然后选择 Commit 提交修改。



### 注意

如果你的分区和示例不一样，挂载到 / 的根分区必须是磁盘上的最后一个分区，以便在启动新虚拟机实例时能够扩展分区大小。同时注意 bsd-cloudinit 目前将假设根分区是磁盘的第二个分区且已硬编码到程序中。

5. 设置root用户密码。

6. 选择 CMOS 时区。

虚拟的CMOS始终存储它的时间时区为UTC,除非你知道原因，否则选择UTC.

7. 选择你当前环境的时区。

8. 从启动时需要启动的服务列表中选择 ssh 。以及可选的其他服务。

9. 可选的，添加用户。

你当前不需要添加用户，如果 freebsd 账户不存在， bsd-cloudinit 程序会添加一个 freebsd 用户账户。用户的ssh秘钥和OpenStack相关联。要定制用户，你可以现在创建，例如，你可能想定制用户的shell。

10. Final config

这个菜单让你更新之前的设置，检查参数是否正确，然后选择 exit 。

11. 在退出后，你可以打开一个shell完成手工配置，选择 Yes 完成一些OpenStack特定的修改：

- a. 设置console:

```
# echo 'console="comconsole,vidconsole"' >> /boot/loader.conf
```

设置console输出到串口控制台，可以由 nova consolelog 查看，以及图形console使用 VNC 或者 Spice 。

- b. 最小启动延时：

```
# echo 'autoboot_delay="1"' >> /boot/loader.conf
```

- c. 下载最新版本的 bsd-cloudinit-installer 。 FreeBSD 10.0 和 9.2的命令不同之处在于fetch 命令处理HTTPS URL不同。

FreeBSD 10.0中 fetch 命令默认价差SSL对端，因此你需要安装 ca\_root\_nss 包，该包包含了根证书并且告诉 fetch 使用，对 FreeBSD 10.0 运行以下命令：

```
# pkg install ca_root_nss
# fetch --ca-cert=/usr/local/share/certs/ca-root-nss.crt
https://raw.github.com/pellaeon/bsd-cloudinit-installer/master/installer.sh
```

FreeBSD 9.2 fetch 不支持https对端认证,FreeBSD 9.2运行以下命令

```
# fetch https://raw.github.com/pellaeon/bsd-cloudinit-installer/master/installer.sh
```

- d. 运行安装程序：

```
# sh ./installer.sh
```

Issue this command to download and install the latest bsd-cloudinit package, and install the necessary prerequisites.

- e. 安装 sudo 并且配置 freebsd 用户无密码访问：

```
# pkg install sudo
# echo 'freebsd ALL=(ALL) NOPASSWD: ALL' > /usr/local/etc/sudoers.d/10-cloudinit
```

12. 关闭系统：

```
# shutdown -s now
```

# 第 6 章 支持创建镜像的工具

Diskimage-builder .....	43
Oz .....	43
VMBuild .....	45
BoxGrinder .....	45
VeeWee .....	45
Packer .....	45
imagefactory .....	45
SUSE Studio .....	45

有几个工具设计为自动创建镜像。

## Diskimage-builder

[diskimage-builder](#) is an automated disk image creation tool that supports a variety of distributions and architectures. Diskimage-builder (DIB) can build images for Fedora, Red Hat Enterprise Linux, Ubuntu, Debian, CentOS, and openSUSE. DIB is organized in a series of elements that build on top of each other to create specific images.

To build an image, call the following script:

```
# disk-image-create ubuntu vm
```

This example creates a generic, bootable Ubuntu image of the latest release.

Further customization could be accomplished by setting environment variables or adding elements to the command-line:

```
# disk-image-create -a armhf ubuntu vm
```

This example creates the image as before, but for arm architecture. More elements are available on [github](#).

## Oz

[Oz](#) is a command-line tool that automates the process of creating a virtual machine image file. Oz is a Python app that interacts with KVM to step through the process of installing a virtual machine. It uses a predefined set of kickstart (Red Hat-based systems) and preseed files (Debian-based systems) for operating systems that it supports, and it can also be used to create Microsoft Windows images. On Fedora, install Oz with yum:

```
# yum install oz
```



### 注意

As of this writing, there are no Oz packages for Ubuntu, so you will need to either install from the source or build your own .deb file.

A full treatment of Oz is beyond the scope of this document, but we will provide an example. You can find additional examples of Oz template files on GitHub at [rackerjoe/oz-image-build/templates](#). Here's how you would create a CentOS 6.4 image with Oz.

Create a template file (we'll call it centos64.tdl) with the following contents. The only entry you will need to change is the <rootpw> contents.

```
<template>
<name>centos64</name>
<os>
<name>CentOS-6</name>
<version>4</version>
<arch>x86_64</arch>
<install type='iso'>
<iso>http://mirror.rackspace.com/CentOS/6/isos/x86_64/CentOS-6.4-x86_64-bin-DVD1.iso</iso>
</install>
<rootpw>CHANGE THIS TO YOUR ROOT PASSWORD</rootpw>
</os>
<description>CentOS 6.4 x86_64</description>
<repositories>
<repository name='epel-6'>
<url>http://download.fedoraproject.org/pub/epel/6/$basearch</url>
<signed>no</signed>
</repository>
</repositories>
<packages>
<package name='epel-release' />
<package name='cloud-utils' />
<package name='cloud-init' />
</packages>
<commands>
<command name='update'>
yum -y update
yum clean all
sed -i '/^HWADDR/d' /etc/sysconfig/network-scripts/ifcfg-eth0
echo -n > /etc/udev/rules.d/70-persistent-net.rules
echo -n > /lib/udev/rules.d/75-persistent-net-generator.rules
</command>
</commands>
</template>
```

This Oz template specifies where to download the Centos 6.4 install ISO. Oz will use the version information to identify which kickstart file to use. In this case, it will be [RHEL6.auto](#). It adds EPEL as a repository and install the epel-release, cloud-utils, and cloud-init packages, as specified in the packages section of the file.

After Oz completes the initial OS install using the kickstart file, it customizes the image with an update. It also removes any reference to the eth0 device that libvirt creates while Oz does the customizing, as specified in the command section of the XML file.

To run this:

```
# oz-install -d3 -u centos64.tdl -x centos64-libvirt.xml
```

- -d3 参数指定 Oz 运行时显示状态信息。
- -u 参数指定 Oz 完成安装后运行自定义的安装动作（安装扩展包，运行其他命令等）
- -x <filename> 参数指定 Oz 输出给libvirt使用的 XML 文件名（否则它默认文件名是诸如 centos64Apr\_03\_2013-12:39:42这种名称）。

If you leave out the `-u` flag, or you want to edit the file to do additional customizations, you can use the `oz-customize` command, using the libvirt XML file that `oz-install` creates. For example:

```
# oz-customize -d3 centos64.tdl centos64-libvirt.xml
```

Oz will invoke libvirt to boot the image inside of KVM, then Oz will ssh into the instance and perform the customizations.

## VMBuilder

[VMBuilder](#) (Virtual Machine Builder) is a command-line tool that creates virtual machine images for different hypervisors. The version of VMBuilder that ships with Ubuntu can only create Ubuntu virtual machine guests. The version of VMBuilder that ships with Debian can create Ubuntu and Debian virtual machine guests.

The [Ubuntu Server Guide](#) has documentation on how to use VMBuilder to create an Ubuntu image.

## BoxGrinder

[BoxGrinder](#) is another tool for creating virtual machine images, which it calls appliances. BoxGrinder can create Fedora, Red Hat Enterprise Linux, or CentOS images. BoxGrinder is currently only supported on Fedora.

## VeeWee

[VeeWee](#) is often used to build [Vagrant](#) boxes, but it can also be used to build KVM images.

## Packer

[Packer](#) is a tool for creating machine images for multiple platforms from a single source configuration.

## imagefactory

[imagefactory](#) is a newer tool designed to automate the building, converting, and uploading images to different cloud providers. It uses Oz as its back-end and includes support for OpenStack-based clouds.

## SUSE Studio

[SUSE Studio](#) is a web application for building and testing software applications in a web browser. It supports the creation of physical, virtual or cloud-based applications and includes support for building images for OpenStack based clouds using SUSE Linux Enterprise and openSUSE as distributions.

# 第 7 章 转换镜像格式

转换一种镜像格式到另一种通常是直接的。

## qemu-img convert: raw, qcow2, QED, VDI, VMDK, VHD

The `qemu-img convert` command can do conversion between multiple formats, including qcow2, QED, raw, VDI (VirtualBox), VHD (Hyper-V), and VMDK (VMware).

表 7.1. `qemu-img` 格式字符列表

镜像格式	<code>qemu-img</code> 参数
qcow2 (KVM, Xen)	qcow2
QED (KVM)	qed
raw	raw
VDI (VirtualBox)	vdi
VHD (Hyper-V)	vpc
VMDK (VMware)	vmdk

This example will convert a raw image file named `centos7.img` to a qcow2 image file.

```
$ qemu-img convert -f raw -O qcow2 centos7.img centos7.qcow2
```

Run the following command to convert a VMDK image file to a raw image file.

```
$ qemu-img convert -f vmdk -O raw centos7.vmdk centos7.img
```

Run the following command to convert a VMDK image file to a qcow2 image file.

```
$ qemu-img convert -f vmdk -O qcow2 centos7.vmdk centos7.qcow2
```



### 注意

The `-f` format flag is optional. If omitted, `qemu-img` will try to infer the image format.

## VBoxManage: VDI (VirtualBox) 转换为 raw

If you've created a VDI image using VirtualBox, you can convert it to raw format using the `VBoxManage` command-line tool that ships with VirtualBox. On Mac OS X, and Linux, VirtualBox stores images by default in the `~/VirtualBox VMs/` directory. The following example creates a raw image in the current directory from a VirtualBox VDI image.

```
$ VBoxManage clonehd ~/VirtualBox\ VMs/fedora21.vdi fedora21.img --format raw
```

# 附录 A. 社区支持

## 目录

文档 .....	47
问答论坛 .....	48
OpenStack 邮件列表 .....	48
OpenStack 维基百科 .....	48
Launchpad的Bug区 .....	48
The OpenStack 在线聊天室频道 .....	49
文档反馈 .....	50
OpenStack分发包 .....	50

以下可用的资源是帮助用户运行和使用OpenStack。OpenStack社区会经常性的改进和增加OpenStack的主要特性，如果用户有问题，请不要在提问题方面犹豫。使用下面列出的资源，以获得OpenStack社区的支持，也能得到一些安装/使用时一些解决问题的思路和方法。

## 文档

For the available OpenStack documentation, see [docs.openstack.org](http://docs.openstack.org).

To provide feedback on documentation, join and use the  
<openstack-docs@lists.openstack.org> mailing list at [OpenStack Documentation Mailing List](#), or [report a bug](#).

以下书籍解释了如何安装一个基于OpenStack云及其相关的组件

- [Installation Guide for openSUSE 13.2 and SUSE Linux Enterprise Server 12](#)
- [Installation Guide for Red Hat Enterprise Linux 7, CentOS 7, and Fedora 21](#)
- [Installation Guide for Ubuntu 14.04 \(LTS\)](#)

以下书籍解释了如何配置和运行一个基于OpenStack的云：

- [架构设计指南](#)
- [云计算平台管理员手册](#)
- [配置参考手册](#)
- [实战指南](#)
- [Networking Guide](#)
- [高可用指南](#)
- [安全指南](#)
- [虚拟机镜像指南](#)

以下书籍解释了如何使用OpenStack图形界面和命令行客户端：

- [应用程序接口快速入门](#)
- [用户指南](#)
- [管理员手册](#)
- [命令行参考](#)

下面文档提供了OpenStack 应用程序接口的参考和向导：

- [OpenStack应用程序接口完全参考\(HTML\)](#)
- [OpenStack应用程序接口完全参考\(PDF\)](#)

The [Training Guides](#) offer software training for cloud administration and management.

## 问答论坛

During the set up or testing of OpenStack, you might have questions about how a specific task is completed or be in a situation where a feature does not work correctly. Use the [ask.openstack.org](https://ask.openstack.org) site to ask questions and get answers. When you visit the <https://ask.openstack.org> site, scan the recently asked questions to see whether your question has already been answered. If not, ask a new question. Be sure to give a clear, concise summary in the title and provide as much detail as possible in the description. Paste in your command output or stack traces, links to screen shots, and any other information which might be useful.

## OpenStack 邮件列表

A great way to get answers and insights is to post your question or problematic scenario to the OpenStack mailing list. You can learn from and help others who might have similar issues. To subscribe or view the archives, go to <http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack>. You might be interested in the other mailing lists for specific projects or development, which you can find [on the wiki](#). A description of all mailing lists is available at <https://wiki.openstack.org/wiki/MailingLists>.

## OpenStack 维基百科

The [OpenStack wiki](#) contains a broad range of topics but some of the information can be difficult to find or is a few pages deep. Fortunately, the wiki search feature enables you to search by title or content. If you search for specific information, such as about networking or OpenStack Compute, you can find a large amount of relevant material. More is being added all the time, so be sure to check back often. You can find the search box in the upper-right corner of any OpenStack wiki page.

## Launchpad的Bug区

The OpenStack community values your set up and testing efforts and wants your feedback. To log a bug, you must sign up for a Launchpad account at <https://launchpad.net/+login>.

You can view existing bugs and report bugs in the Launchpad Bugs area. Use the search feature to determine whether the bug has already been reported or already been fixed. If it still seems like your bug is unreported, fill out a bug report.

一些小贴士：

- 提供清晰、简洁的语法。
- 尽可能提供详细的细节描述。将命令行的输出或者trace输出粘贴出来，如果是截图请贴链接，以及其他任何有用的信息。
- 确保包含了软件和包的版本信息，尤其是使用的正在开发中的分支，诸如”Juno release” vs git commit bc79c3ecc55929bac585d04a03475b72e06a3208，这样的描述。
- 任何特别的部署信息都是有用的。例如用户使用的是Ubuntu 14.04，或者多节点安装。

以下列出Launchpad Bug区：

- [Bugs: OpenStack 块存储 \(cinder\)](#)
- [Bugs: OpenStack 计算 \(nova\)](#)
- [Bugs: OpenStack 仪表盘 \(horizon\)](#)
- [Bugs: OpenStack 认证 \(keystone\)](#)
- [Bugs: OpenStack Image service \(glance\)](#)
- [Bugs: OpenStack 网络 \(neutron\)](#)
- [Bugs: OpenStack 对象存储 \(swift\)](#)
- [Bugs: 裸金属服务 \(ironic\)](#)
- [Bugs: 数据处理服务 \(sahara\)](#)
- [Bugs: Database service \(trove\)](#)
- [Bugs: 编排 \(heat\)](#)
- [Bugs: 计量 \(ceilometer\)](#)
- [Bugs: 消息服务 \(zaqar\)](#)
- [Bugs: OpenStack 应用程序接口文档 \(developer.openstack.org\)](#)
- [Bugs: OpenStack 文档 \(docs.openstack.org\)](#)

## The OpenStack 在线聊天室频道

The OpenStack community lives in the #openstack IRC channel on the Freenode network. You can hang out, ask questions, or get immediate feedback for urgent and pressing issues. To install an IRC client or use a browser-based client, go to <https://webchat.freenode.net/>. You can also use Colloquy (Mac OS X, <http://colloquy.info/>), mIRC (Windows, <http://>

[www.mirc.com/](http://www.mirc.com/)), or XChat (Linux). When you are in the IRC channel and want to share code or command output, the generally accepted method is to use a Paste Bin. The OpenStack project has one at <http://paste.openstack.org>. Just paste your longer amounts of text or logs in the web form and you get a URL that you can paste into the channel. The OpenStack IRC channel is #openstack on irc.freenode.net. You can find a list of all OpenStack IRC channels at <https://wiki.openstack.org/wiki/IRC>.

## 文档反馈

To provide feedback on documentation, join and use the <[openstack-docs@lists.openstack.org](mailto:openstack-docs@lists.openstack.org)> mailing list at [OpenStack Documentation Mailing List](#), or [report a bug](#).

## OpenStack分发包

以下是Linux发行版针对OpenStack的社区支持：

- Debian: <https://wiki.debian.org/OpenStack>
- CentOS, Fedora, 以及 Red Hat Enterprise Linux: <https://www.rdoproject.org/>
- openSUSE 和 SUSE Linux Enterprise Server: <https://en.opensuse.org/Portal:OpenStack>
- Ubuntu: [ubuntu官方服务器团队之OpenStack云](#)