

# Fedora 12

## 虚拟化指南

Fedora 中虚拟化的权威指南



Christoph Curran

# Fedora 12 虚拟化指南

## Fedora 中虚拟化的权威指南

### 版 1

作者

Christoph Curran

[ccurran@redhat.com](mailto:ccurran@redhat.com)

Copyright © 2009 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution—Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at <http://creativecommons.org/licenses/by-sa/3.0/>. The original authors of this document, and Red Hat, designate the Fedora Project as the "Attribution Party" for purposes of CC-BY-SA. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

For guidelines on the permitted uses of the Fedora trademarks, refer to [https://fedoraproject.org/wiki/Legal:Trademark\\_guidelines](https://fedoraproject.org/wiki/Legal:Trademark_guidelines).

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

All other trademarks are the property of their respective owners.

Fedora 12 虚拟化指南包含在 Fedora 12 中使用的有关安装、配置、管理、提示、陷阱以及故障排除等虚拟化技术信息。

---

序言	vii
1. 关于	vii
2. 文档约定	vii
2.1. 排版约定	vii
2.2. 抬升式引用约定	ix
2.3. 备注及警告	ix
3. 我们需要您的反馈！	x
I. Installation	1
1. 安装虚拟化软件包	3
1.1. 在全新 Fedora 安装过程中安装 KVM	3
1.2. 在现有 Fedora 系统中安装 KVM 软件包	5
2. 虚拟客户端安装总览	7
2.1. 使用 virt-install 创建客户端	7
2.2. 使用 virt-manager 创建客户端	7
2.3. 使用 PXE 安装客户端	16
3. 客户端操作系统安装过程	23
3.1. 将红帽企业版 Linux 5 作为半虚拟客户端安装	23
3.2. 将红帽企业版 Linux 作为全虚拟客户端安装	64
3.3. 将 Windows XP 作为全虚拟客户端安装	73
3.4. 将 Windows 服务器 2003 作为全虚拟客户端安装	90
3.5. 将 Windows 服务器 2008 作为全虚拟客户端安装	93
II. Configuration	105
4. 虚拟块设备	107
4.1. 创建一个虚拟软盘控制程序	107
4.2. 在客户端中添加存储设备	108
4.3. 配置持久存储	111
4.4. 在客户端中添加虚拟光驱或者 DVD 设备	113
5. 共享存储和虚拟化	115
5.1. 使用 iSCSI 存储客户端	115
5.2. 使用 NFS 存储客户端	115
5.3. 使用 GFS2 存储客户端	115
6. 服务器最佳实践	117
7. 虚拟化的安全性	119
7.1. SELinux 和虚拟化	119
7.2. SELinux 的相关事宜	120
8. 网络配置	121
8.1. libvirt 的网络地址转换 (NAT)	121
8.2. 使用 libvirt 桥接的联网	122
9. KVM 半虚拟驱动程序	125
9.1. 安装 KVM Windows 半虚拟驱动程序	125
III. Administration	133
10. 用 xend 管理客户端	135

11. KVM 客户端计时管理	137
12. KVM 实时迁移	139
12.1. 动态迁移要求	139
12.2. 共享存储示例：使用 NFS 进行简单迁移	140
12.3. 使用 virsh 进行实时 KVM 迁移	141
12.4. 使用 virt-manager 迁移	142
13. 虚拟客户端的远程管理	153
13.1. 使用 SSL 进行远程管理	153
13.2. 使用 TLS 和 SSL 进行远程管理	154
13.3. 传输模式	155
IV. 虚拟化参考指南	159
14. 虚拟化工具	161
15. 使用 virsh 管理客户端	163
16. 用虚拟机管理器（virt-manager）管理客户端	173
16.1. 打开连接窗口	173
16.2. 虚拟机管理器主窗口	174
16.3. 虚拟机管理器详情窗口	175
16.4. 虚拟机图形控制台	176
16.5. Starting virt-manager	177
16.6. 恢复保存的机器	178
16.7. 显示客户端详情	179
16.8. 状态监控	184
16.9. 显示客户端识别符	185
16.10. 显示客户端状态	186
16.11. 显示虚拟 CPU	187
16.12. 显示 CPU 用量	188
16.13. 显示内存用量	189
16.14. 管理虚拟网络	190
16.15. 创建虚拟网络	192
V. Tips and Tricks	201
17. 提示及陷阱	203
17.1. 自动启动客户端	203
17.2. 在 KVM 和 Xen 管理程序间互换	203
17.2.1. Xen 到 KVM	203
17.2.2. KVM 到 Xen	205
17.3. 使用 qemu-img	206
17.4. KVM 中的过量使用	207
17.5. 修改 /etc/grub.conf	208
17.6. 验证虚拟化扩展	209
17.7. 鉴别客户端类型及使用	210
17.8. 生成新的唯一 MAC 地址	211
17.9. 非常安全的 ftpd	212
17.10. 配置 LUN 持久性	212
17.11. 为客户端禁用 SMART 磁盘监控	214
17.12. 克隆客户端配置文件	214
17.13. 复制现有客户端及其配置文件	214

---

18. 生成自定义 libvirt 脚本	217
18.1. 使用带 virsh 的 XML 配置文件	217
VI. Troubleshooting	219
19. Troubleshooting	221
19.1. 回送设备错误	221
19.2. 在 BIOS 中启用 Intel VT 和 AMD-V 虚拟化硬件扩展	221
A. 附加资源	223
A.1. 在线资源	223
A.2. 安装的文档	223
B. 修订记录	225
C. 版权	227
术语表	229

---

---

# 序言

本书是 Fedora 12 虚拟化指南。这个指南覆盖了在 Fedora 12 中使用和管理虚拟化的各个方面。

## 1.0 关于

本书分为七个部分：

- 系统要求
- Installation
- Configuration
- Administration
- 参考
- Tips and Tricks
- Troubleshooting

## 2.0 文档约定

本手册使用几个约定来突出某些用词和短语以及信息的某些片段。

在 PDF 版本以及纸版中，本手册使用在 *Liberation* <sup>[1]</sup> 套件中选出的字体。如果您在您的系统中安装了 Liberation 字体套件，它还可用于 HTML 版本。如果没有安装，则会显示可替换的类似字体。请注意：红帽企业 Linux 5 以及其后的版本默认包含 Liberation 字体套件。

### 2.1.0 排版约定

我们使用四种排版约定突出特定用词和短语。这些约定及其使用环境如下。

固定粗体

用来突出系统输入，其中包括 shell 命令、文件名以及路径。还可用来突出按键以及组合键。例如：

To see the contents of the file `my_next_bestselling_novel` in your current working directory, enter the `cat my_next_bestselling_novel` command at the shell prompt and press Enter to execute the command.

The above includes a file name, a shell command and a key cap, all presented in Mono-spaced Bold and all distinguishable thanks to context.

Key-combinations can be distinguished from key caps by the hyphen connecting each part of a key-combination. For example:

Press Enter to execute the command.

Press Ctrl+Alt+F1 to switch to the first virtual terminal. Press Ctrl+Alt+F7 to return to your X-Windows session.

---

<sup>1</sup> <https://fedorahosted.org/liberation-fonts/>

The first sentence highlights the particular key cap to press. The second highlights two sets of three key caps, each set pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in Mono-spaced Bold. For example:

File-related classes include `filesystem` for file systems, `file` for files, and `dir` for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialogue box text; labelled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose System > Preferences > Mouse from the main menu bar to launch Mouse Preferences. In the Buttons tab, click the Left-handed mouse check box and click Close to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a gedit file, choose Applications > Accessories > Character Map from the main menu bar. Next, choose Search > Find... from the Character Map menu bar, type the name of the character in the Search field and click Next. The character you sought will be highlighted in the Character Table. Double-click this highlighted character to place it in the Text to copy field and then click the Copy button. Now switch back to your document and choose Edit > Paste from the gedit menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in Proportional Bold and all distinguishable by context.

Note the > shorthand used to indicate traversal through a menu and its sub-menus. This is to avoid the difficult-to-follow 'Select Mouse from the Preferences sub-menu in the System menu of the main menu bar' approach.

### ***Mono-spaced Bold Italic*** or ***Proportional Bold Italic***

Whether Mono-spaced Bold or Proportional Bold, the addition of Italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type ssh ***username@domain.name*** at a shell prompt. If the remote machine is example.com and your username on that machine is john, type ssh john@example.com.

The mount -o remount ***file-system*** command remounts the named file system. For example, to remount the /home file system, the command is mount -o remount /home.

To see the version of a currently installed package, use the rpm -q ***package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.



Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

When the Apache HTTP Server accepts requests, it dispatches child processes or threads to handle them. This group of child processes or threads is known as a *server-pool*. Under Apache HTTP Server 2.0, the responsibility for creating and maintaining these server-pools has been abstracted to a group of modules called *Multi-Processing Modules (MPMs)*. Unlike other modules, only one module from the MPM group can be loaded by the Apache HTTP Server.

## 2.2. 抬升式引用约定

二，一般是多行，数据输入要明显与周围文本分开。

将发送到终端的输入设置为 Mono-spaced Roman 并显示为：

```
books Desktop documentation drafts mss photos stuff svn
books_tests Desktop1 downloads images notes scripts svgs
```

源码列表也设为 Mono-spaced Roman，但以下面的形式出现和突出：

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object ref = iniCtx.lookup("EchoBean");
        EchoHome home = (EchoHome) ref;
        Echo echo = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

## 2.3. 备注及警告

最后，我们使用三种视觉形式来突出那些可能被忽视的信息。



### 备注

备注是对手头任务的提示、捷径或者另外的解决方法。忽略提示不会造成负面后果，但您可能会错过一个更省事的诀窍。



### 重要

重要框中的内容是那些容易错过的事情：配置更改只可用于当前会话，或者在应用更新前要重启服务。忽略重要框中的内容不会造成数据丢失但可能会让您抓狂。



### 警告

警告是不应被忽略的。忽略警告信息很可能导致数据丢失。

## 3. 我们需要您的反馈！

如果你发现了排印错误或者有改进本文档的建议，我们希望能听取你的意见。请在 Bugzilla <http://bugzilla.redhat.com/bugzilla/> 里提交关于产品 Fedora Documentation. 的报告。

提交报告时，请记得提及文档的标识符：*Virtualization\_Guide*

如果你有改进本文档的建议，请尽量将其明确化。如果你发现了错误，请指出章节号以及其周围的相关文字，以便我们尽快找到并更正该错误。

---

## 部分 I. Installation

### 虛擬化的安裝主題

这些章节论述了如何设置主机以及安装 Fedora 虚拟客户端。建议您仔细阅读这些章节以确保成功安装虚拟客户端操作系统。

---

---

# 安装虚拟化软件包

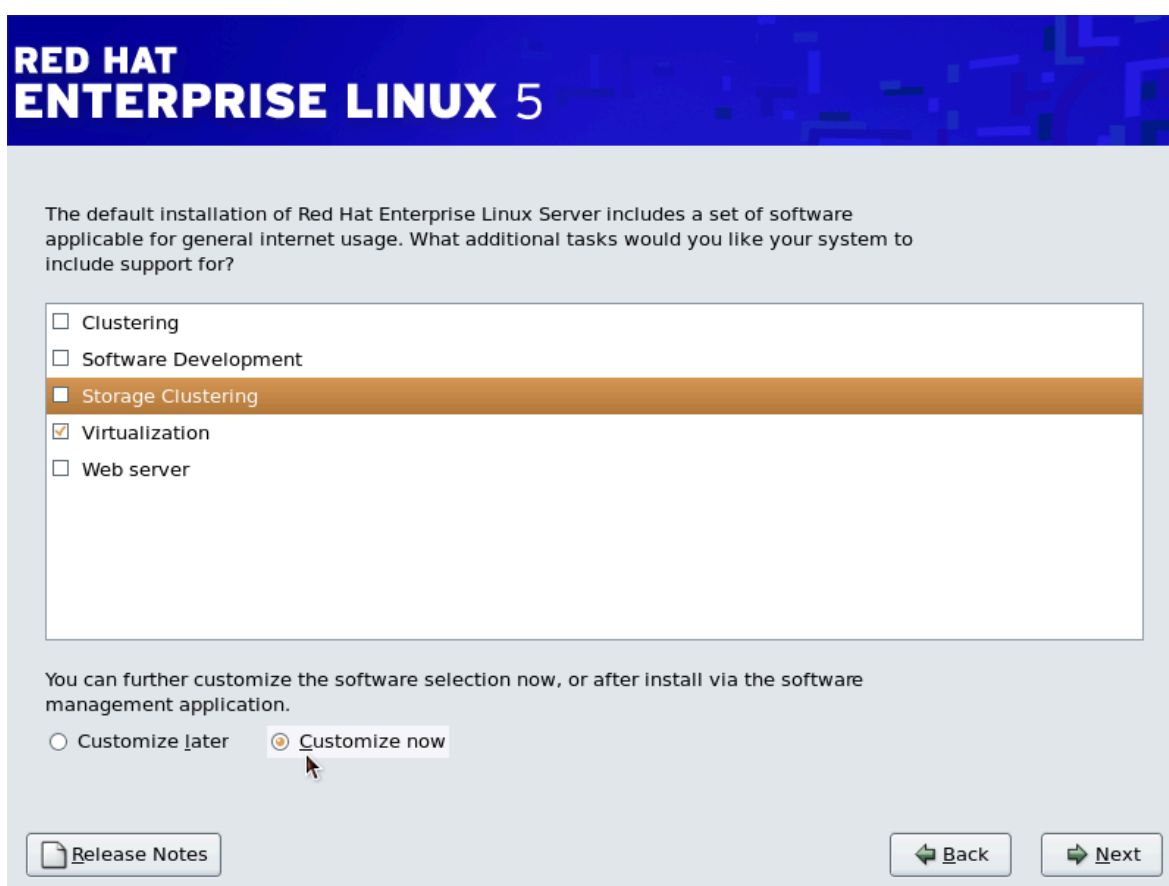
## 1.1. 在全新 Fedora 安装过程中安装 KVM

这部分论述了在全新 Fedora 安装过程中安装虚拟化工具以及 KVM 软件包。



安装过程中需要帮助吗？  
□□□□□□ (详见 [redhat.com](http://redhat.com)<sup>1</sup>) 中详细论述了 Fedora 12 的安装。

1. 从 Fedora 12 安装 CD-ROM、DVD 或者 PXE 启动互动 Fedora 安装。
2. 完成软件包选择步骤前的步骤。

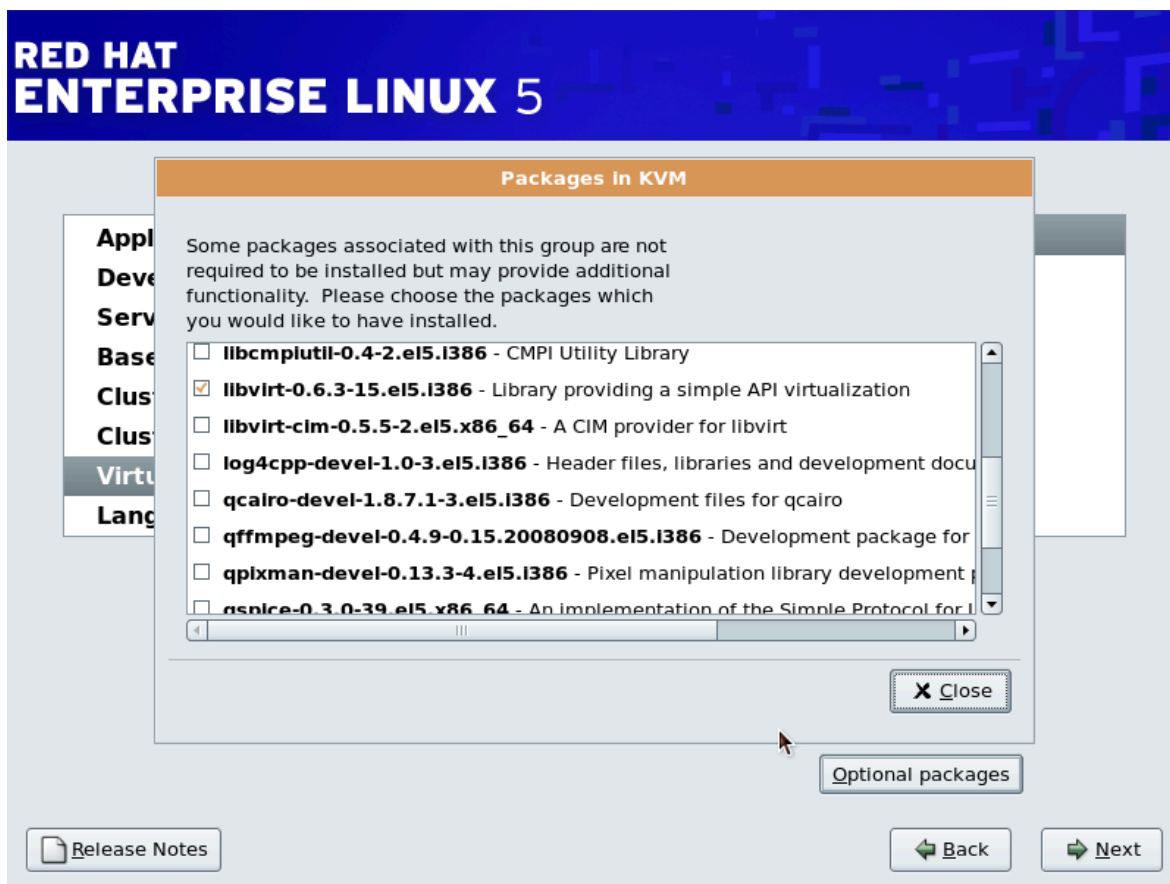


选择「虚拟化」软件包组群以及「现在定制」单选按钮。

3. 请选择「KVM」软件包组。取消对「虚拟化」软件包组的选择。这样会选择 KVM 管理程序、virt-manager、libvirt 和 virt-viewer 进行安装。



- 定制软件包（如果需要）  
如果您需要其它虚拟化软件包则请定制「虚拟化」组。



按「下一步」旁边的「关闭」继续安装。

### 使用 kickstart 文件安装 KVM 软件包

这部分论述了如何使用 Kickstart 文件安装带 KVM 管理程序软件包的 Fedora。Kickstart 文件允许大量的自动安装，从而无需为每个独立的系统进行用户手动安装。这部分讨论的步骤将帮助您创建并使用 Kickstart 文件来安装含有虚拟化软件包的 Fedora。

在您 Kickstart 文件的 `%packages` 部分附加以下软件包组：

```
%packages
@kvm
```

Kickstart 文件详情请参考 [Fedora 12](http://docs.fedoraproject.org) 中的 Fedora 项目网页 <http://docs.fedoraproject.org>。

## 1.2. 在现有 Fedora 系统中安装 KVM 软件包

这部分论述了要在正常工作的 Fedora 12 系统中安装 KVM 管理程序的步骤。

### 使用 yum 安装 KVM 管理程序

要在 Fedora 中使用虚拟化，您需要有 `kvm` 软件包。`kvm` 软件包中含有 KVM 内核模块，它可在默认 Linux 内核中提供 KVM 管理程序。

要安装 `kvm` 软件包，请运行：

```
# yum install kvm
```

现在安装附加虚拟化管理软件包。

推荐的虚拟化软件包：

`python-virtinst`

提供创建虚拟机的 `virt-install` 命令。

`libvirt`

`libvirt` 是一个可与管理程序互动的 API 程序库。`libvirt` 使用 `xm` 虚拟化构架以及 `virsh` 命令行工具管理和控制虚拟机。

`libvirt-python`

`libvirt-python` 软件包中含有一个模块，它允许使用由 Python 编程语言编写的应用程序使用由 `libvirt` API 提供的界面。

`virt-manager`

`virt-manager`，也称 Virtual Machine Manager，它可为管理虚拟机提供图形工具。它使用 `libvirt` 程序库作为管理 API。

安装其它推荐的虚拟化软件包：

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



# 虚拟客户端安装总览

您在主机系统中安装虚拟化软件包后即可创建客户端操作系统。本章论述了在虚拟机中安装客户端操作系统的一般步骤。您可以在 `virt-manager` 中使用 `新建` 按钮创建客户端，也可以使用命令行界面 `virt-install` 进行此操作。本章论述了这两种方法。

红帽企业版 Linux 的每个具体版本都有详细的安装说明，其它 Linux 发行本、Solaris 和 Windows 的说明请参考 [0 3 0 000000000000](#)。

## 2.1.0 使用 `virt-install` 创建客户端

您可以使用 `virt-install` 命令从命令行创建虚拟客户端。`virt-install` 可以交互方式使用，也可以使用脚本从而自动生成虚拟机。将 Kickstart 文件与 `virt-install` 共同使用即可进行虚拟机的自动安装。

`virt-install` 工具提供很多选项，它们可提供该命令行。要查看选项完整列表请运行：

```
$ virt-install --help
```

`virt-install` man page 还记录了每个命令选项和重要变量。

`qemu-img` 是一个关联命令，可在 `virt-install` 前用来配置存储选项。

`--vnc` 是一个重要选项，它可为客户端安装打开图形窗口。

这个示例在 CE-ROM 中使用虚拟联网和 5GB 基于文件的块设备映像创建了名为 `rhel13support` 的红帽企业版 Linux 3 客户端。这个示例使用的是 KVM 管理程序。

```
# virt-install --accelerate --hvm --connect qemu:///system \  
  --network network:default \  
  --name rhel13support --ram=756 \  
  --file=/var/lib/libvirt/images/rhel13support.img \  
  --file-size=6 --vnc --cdrom=/dev/sr0
```

### 例02.1.0 使用 `virt-install` 和 KVM 创建红帽企业版 Linux 3 客户端

```
# virt-install --name Fedora11 --ram 512 --file=/var/lib/libvirt/images/Fedora11.img \  
  --file-size=3 --vnc --cdrom=/var/lib/libvirt/images/Fedora11.iso
```

### 例02.2.0 使用 `virt-install` 创建 Fedora 11 客户端

## 2.2.0 使用 `virt-manager` 创建客户端

`virt-manager` 即虚拟机管理器，是创建和管理虚拟客户端的图形工具。

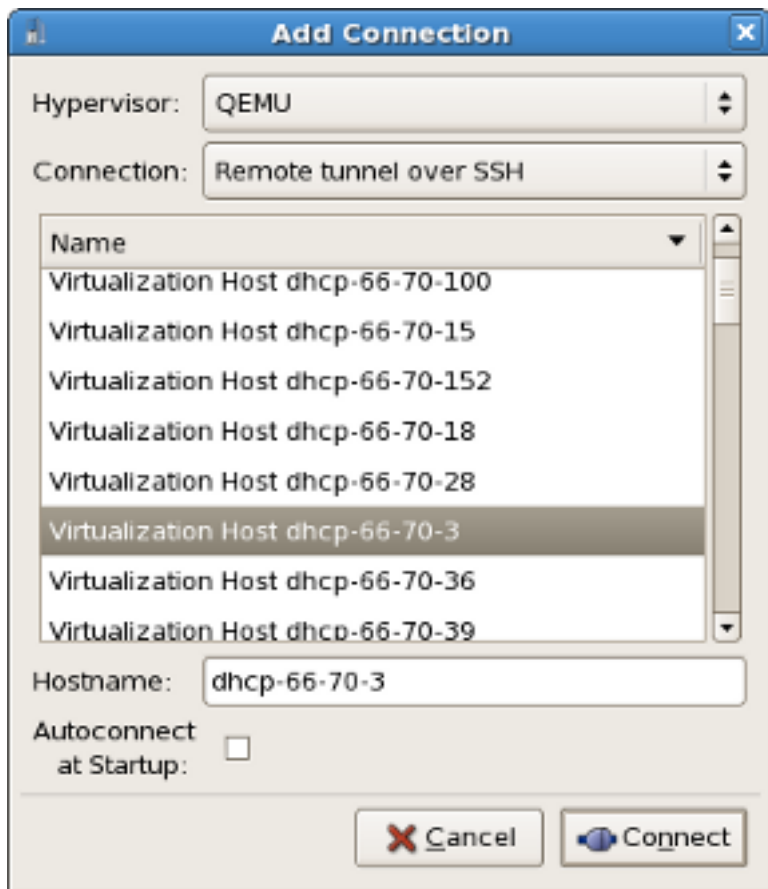
### 过程02.1.0 使用 `virt-manager` 创建虚拟客户端

1. 要启动 `virt-manager`，请以根用户身份运行以下命令：

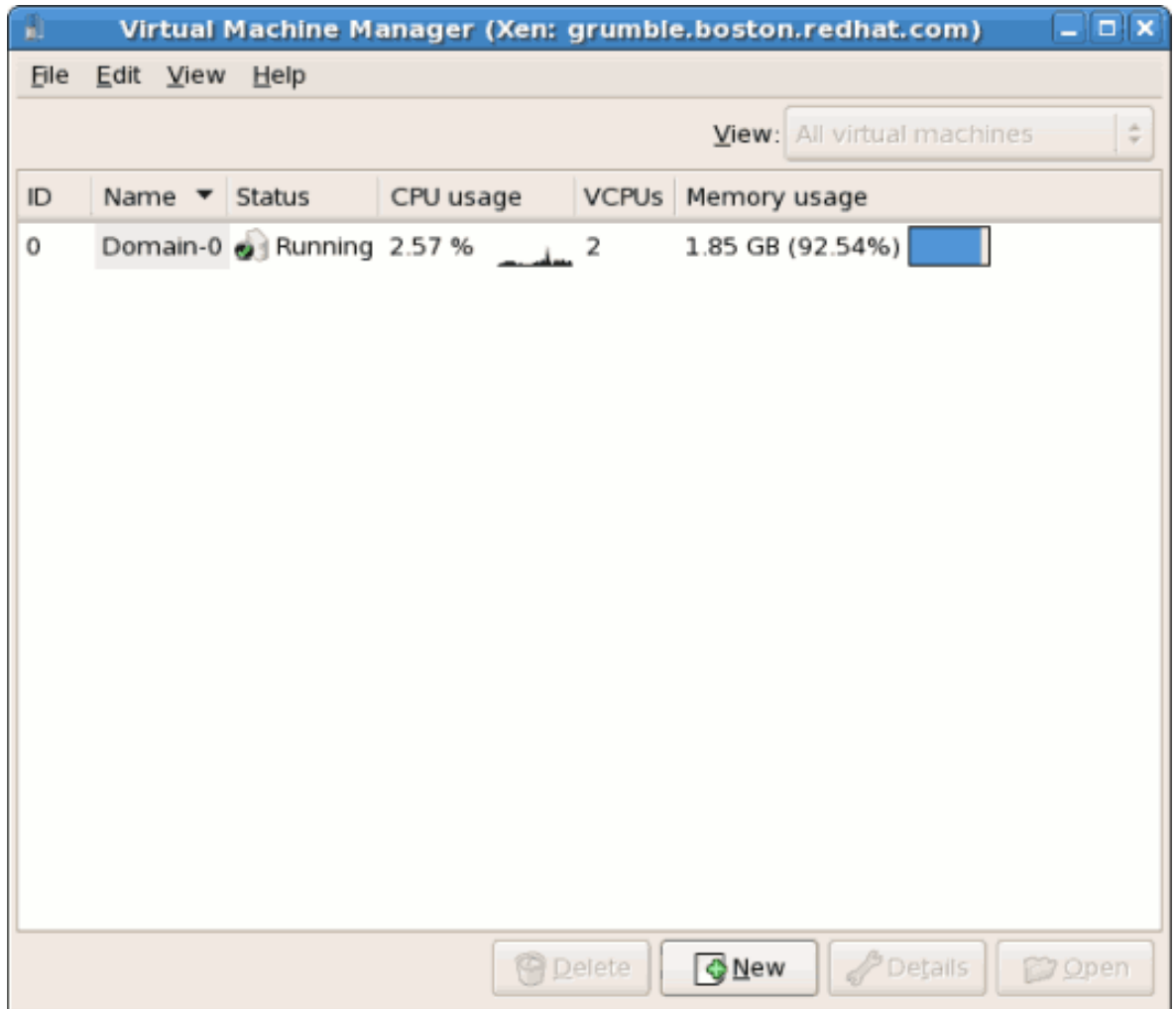
```
# virt-manager &
```

`virt-manager` 会打开图形用户界面窗口。没有根特权或者没有配置 `sudo` 的用户无法使用某些功能，其中包括 `新建` 按钮，就是说您无法创建新的虚拟客户端。

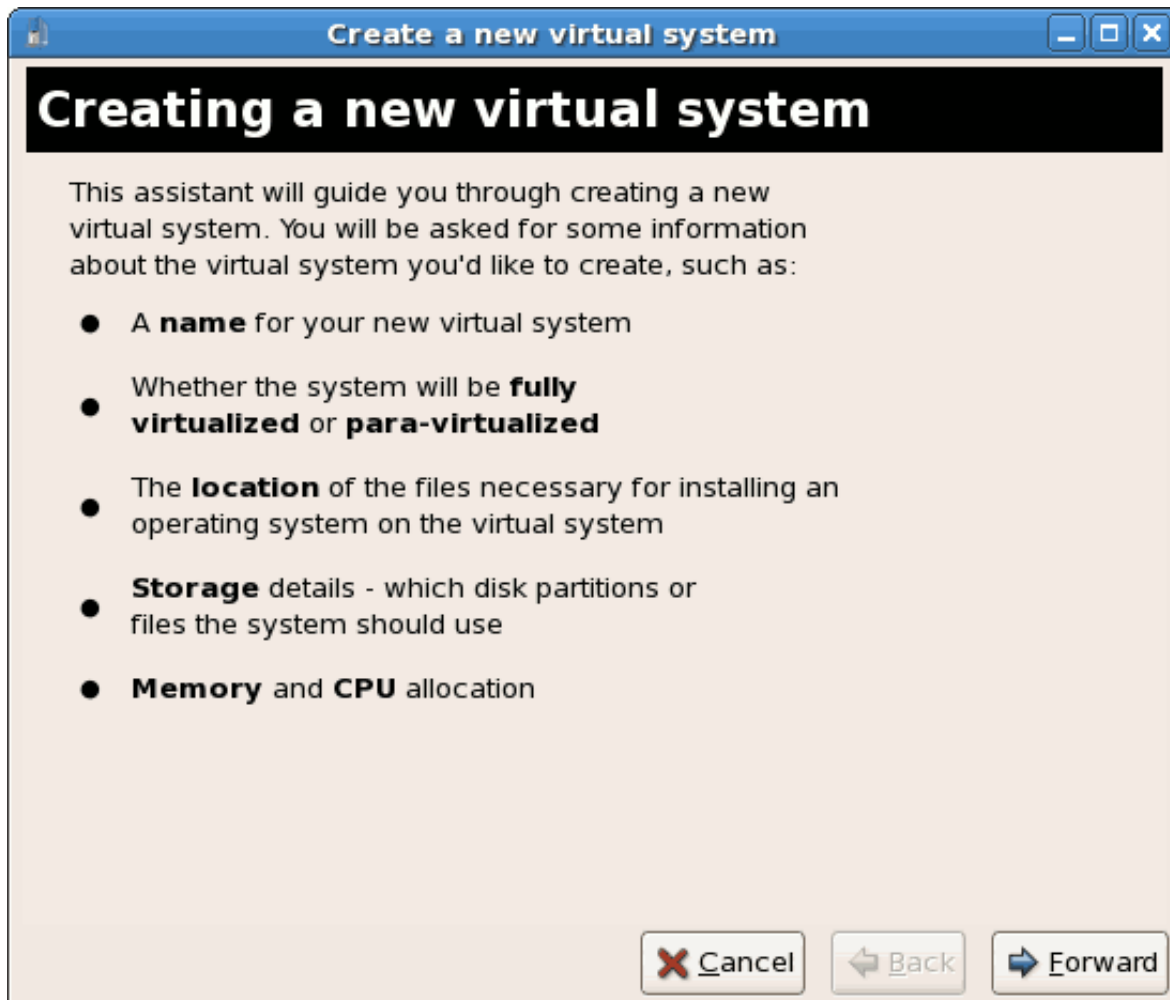
2. 打开「文件 > 打开连接」，则会出现下面的对话框。选择监控程序并点击 连接 按钮：



3. virt-manager 窗口可允许您创建新虚拟机。点击 新建 按钮创建新的客户端。这会打开如下截屏中所示的向导。



4. 「创建新虚拟系统」窗口提供了您要创建虚拟机所必须提供的信息总结：

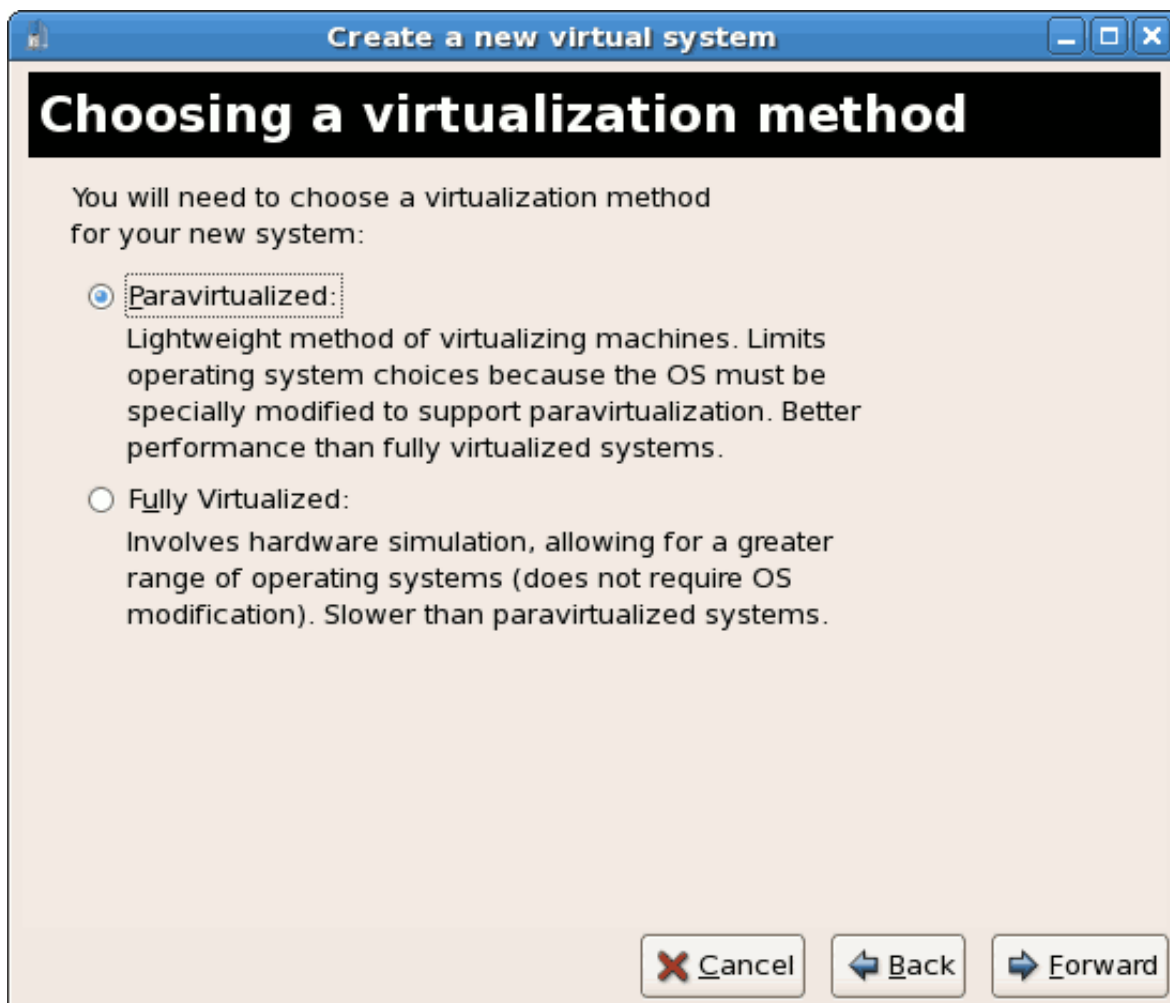


为您的安装复查该信息，并点击 前进 按钮。

5. 此时会出现「选择虚拟化方法」窗口。选择「半虚拟」或者「全虚拟」。

全虚拟化需要系统使用 Intel® VT 或者 AMD-V 处理器。如果没有出现虚拟化扩展，那么就无法选择「全虚拟」或者「启用内核/硬件加速」按钮。如果目前使用的不是 kernel-xen 内核，则「半虚拟」选项就会是灰色的。

如果您连接到 KVM 管理程序，则只能使用全虚拟化。

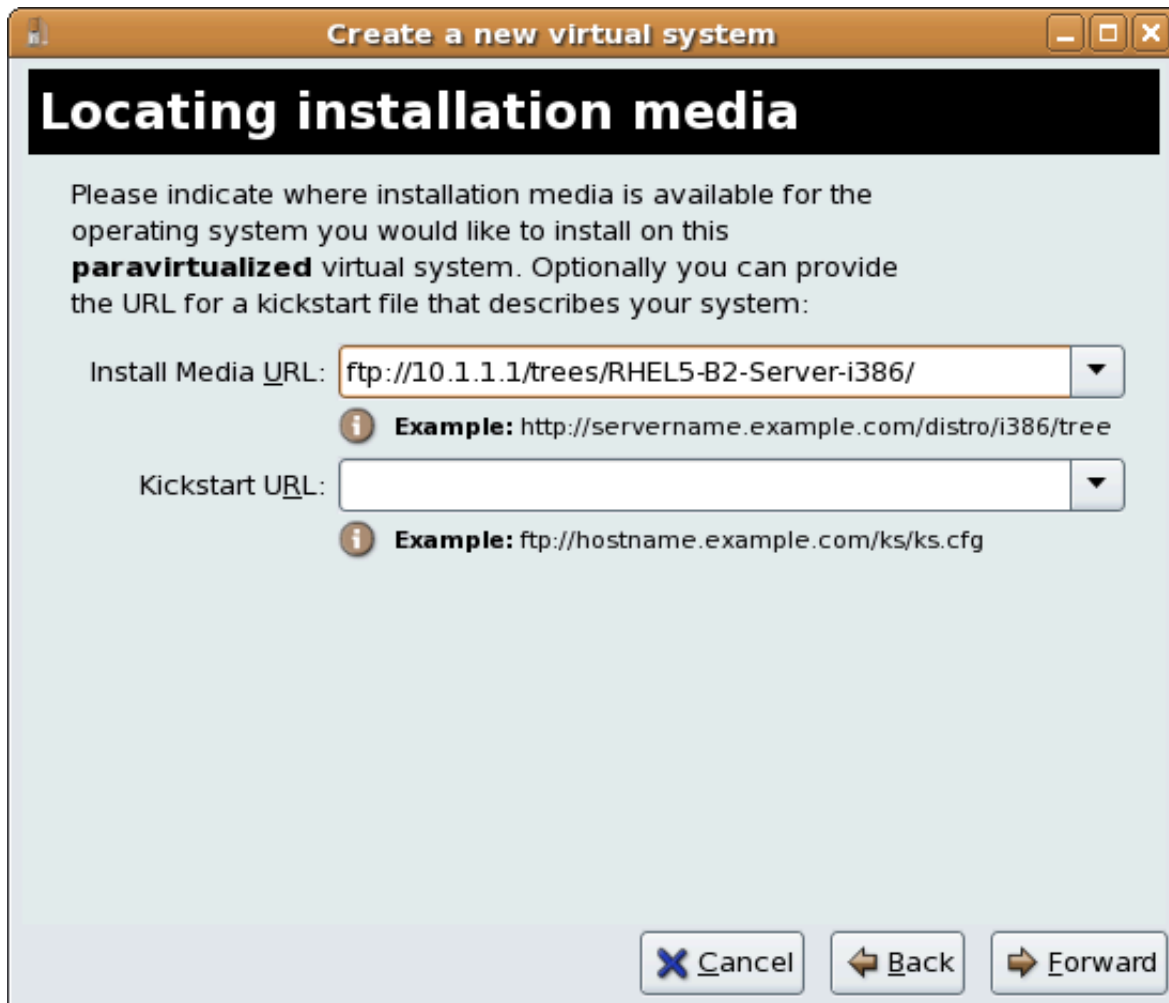


选择虚拟化类型并点击 下一步 按钮。

6. 「定位安装介质」提示您选择安装类型的安装介质。这一步依赖您前一步的选择。
  - a. 半虚拟安装需要使用以下网络协议之一访问安装树：HTTP、FTP 或者 NFS。安装介质 URL 必须含有 Fedora 安装树。这个安装树是使用 NFS、FTP 或者 HTTP 托管。可使用主机中的网络服务或者另一个镜像托管网络服务和文件。

使用 CE-ROM 或者 DVD 映像（标记为 .iso 文件），挂载 CD-ROM 映像并使用以上协议之一托管挂载的文件。

还可以从 Fedora 镜像中复制安装树。



- b. 全虚拟客户端安装需要在本地有可引导安装 DVD、CD-ROM 或者可引导安装 DVD 或者 CD-ROM 的映像（即 .iso 或者 .img 文件类型）。Windows 安装使用 DVD、CD-ROM 或者 .iso 文件。很多 Linux 和类似 UNIX 操作系统在完成通过基于安装树进行的网络安装前要使用 .iso 文件安装基本系统。



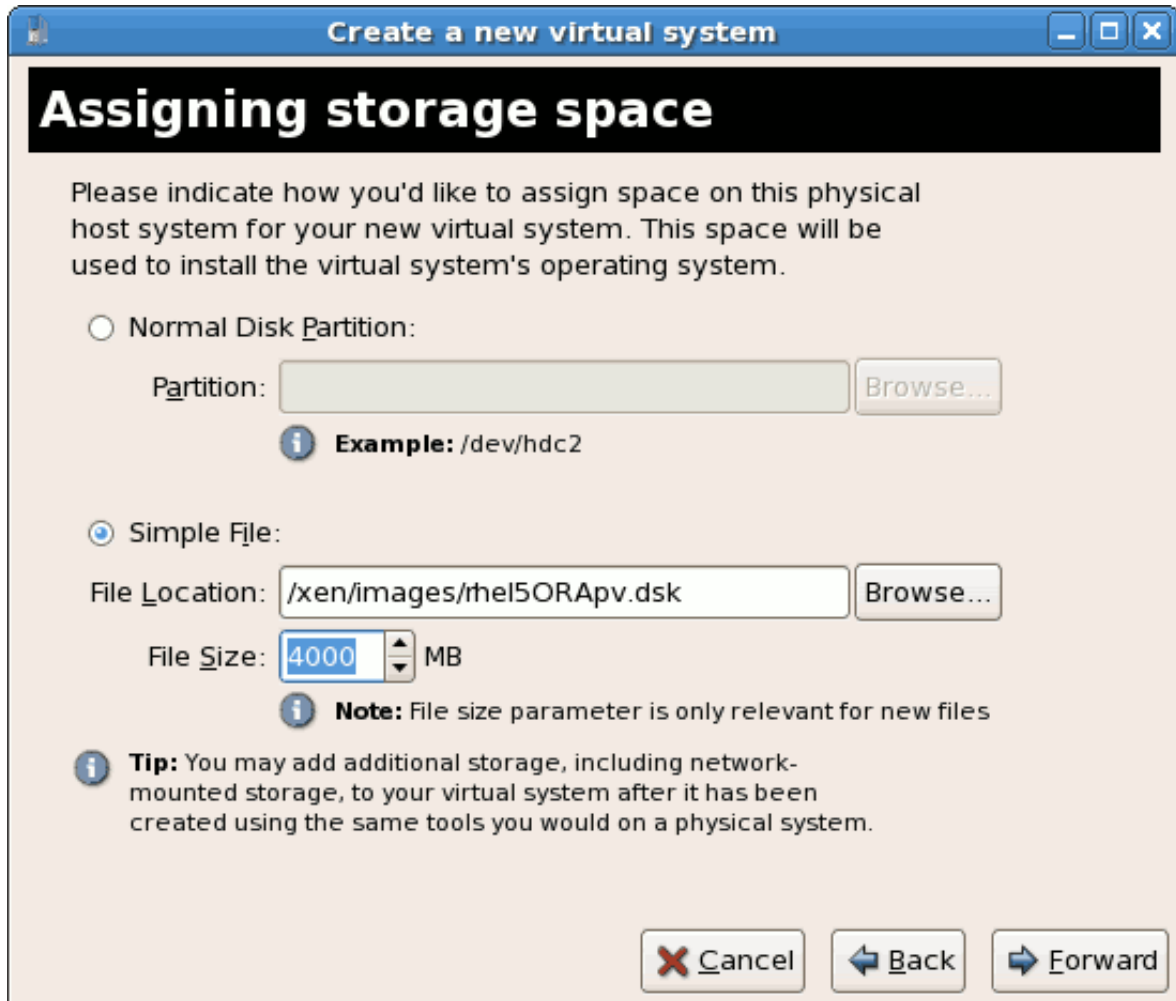
选择正确的安装介质后，点击 前进 按钮。

7. The Assigning storage space window displays. Choose a disk partition, LUN or create a file based image for the guest storage.

在 Fedora 中对基于文件映像的规定是所有基于文件的客户端映像都应位于 `/var/lib/xen/images/` 目录中。SELinux 禁止在其它任何位置保存基于文件的映像。如果您以 enforcing 模式运行 SELinux，则请在 [7.1 “SELinux 配置”](#) 参考安装客户端的详情。

Your guest storage image should be larger than the size of the installation, any additional packages and applications, and the size of the guests swap file. The installation process will choose the size of the guest's swap file based on size of the RAM allocated to the guest.

Allocate extra space if the guest needs additional space for applications or other data. For example, web servers require additional space for log files.



Choose the appropriate size for the guest on your selected storage type and click the Forward button.



**注意**

建议您使用默认目录保存虚拟机映像，默认目录为 `/var/lib/xen/images/`。如果您要使用不同的位置（比如在本示例中使用 `/xen/images/`），请确定将其添加到 SELinux 策略中并在您继续安装前对其进行重新标记（后面会介绍如何修改 SELinux 策略）。

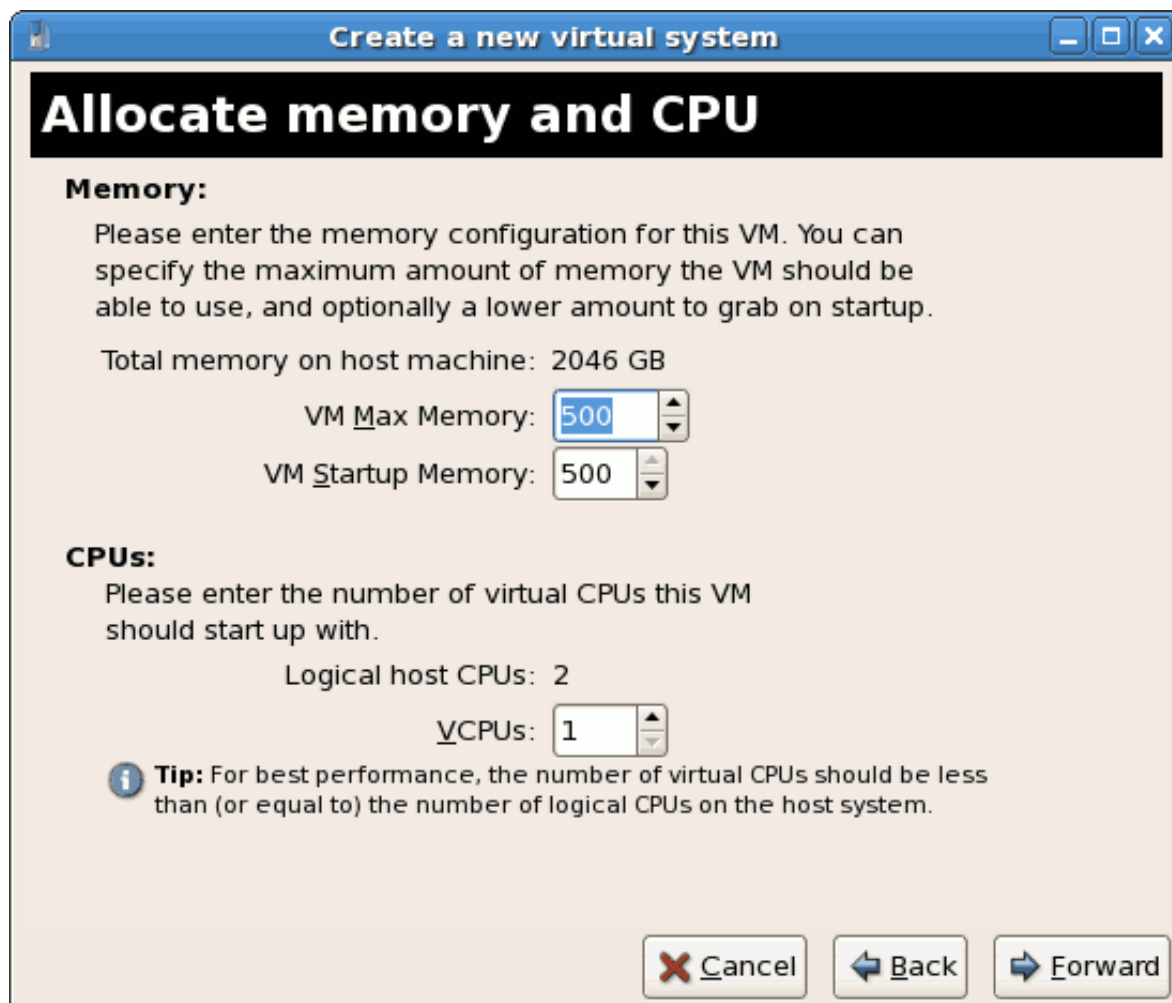
- The Allocate memory and CPU window displays. Choose appropriate values for the virtualized CPUs and RAM allocation. These values affect the host's and guest's performance.

客户端需要足够的物理内存（RAM）方可有效运行。请选择适合您客户端操作系统和应用程序要求的内存值。大多数操作系统需要至少 512MB RAM 方可有效工作。请记住：客户端使用的是物理 RAM。运行客户端过多或者没有为主机系统留下足够内存将导致大量使用虚拟内存。虚拟内存非常缓慢，将导致系统性能和反应能力大大降低。请确定您为所有主机和客户端分配了足够内存以便其有效工作。

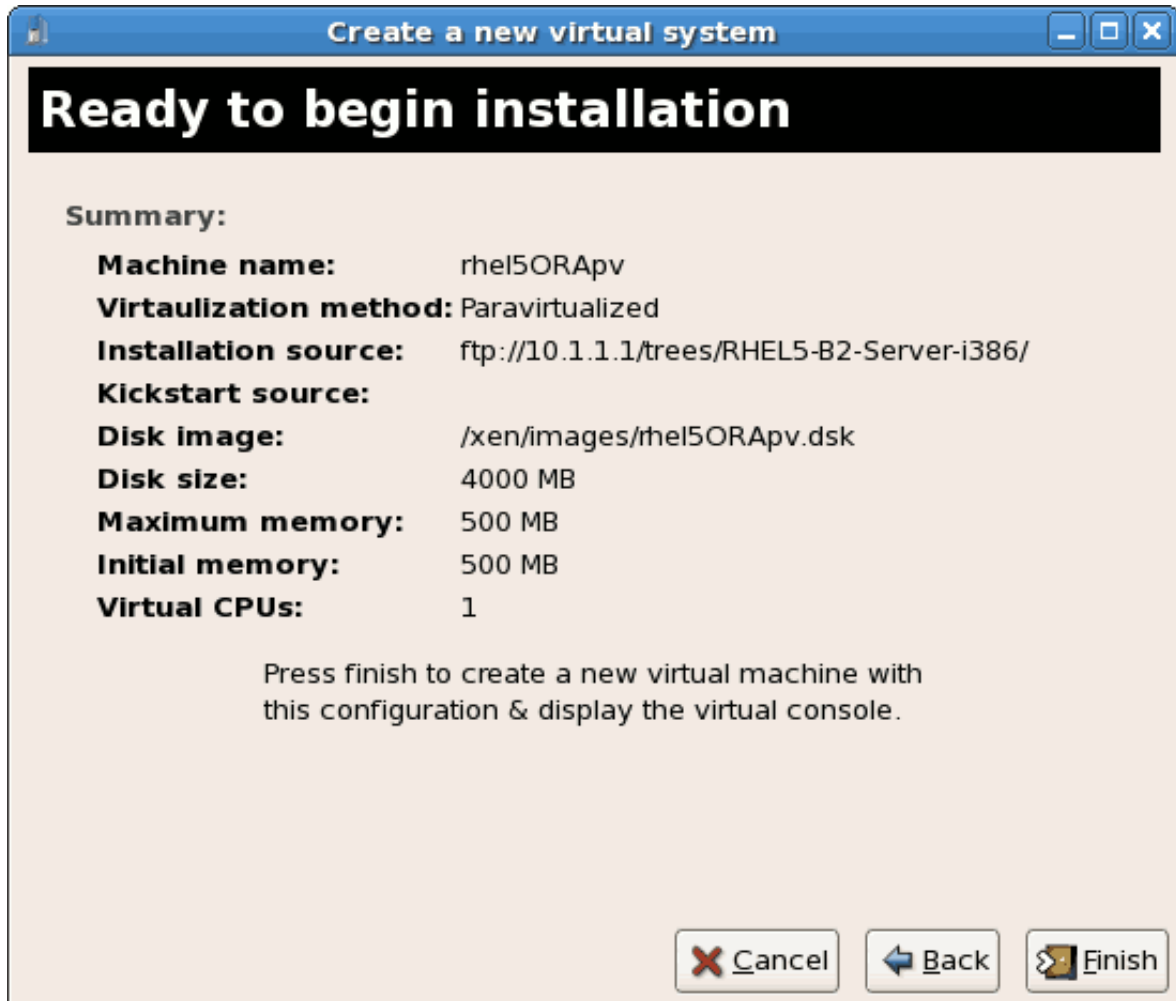
Assign sufficient virtual CPUs for the virtualized guest. If the guest runs a multithreaded application assign the number of virtualized CPUs it requires to run most efficiently. Do not assign more virtual CPUs than there are physical processors (or hyper-threads) available on the host



system. It is possible to over allocate virtual processors, however, over allocating has a significant, negative affect on guest and host performance due to processor context switching overheads.



9. 准备安装窗口显示您输入的所有配置信息总揽。检查这些信息，并在需要使用 返回 按钮进行更改。对所有数据满意后请点击 完成 按钮，则会开始安装过程：



打开的 VNC 窗口显示客户端操作系统安装过程

这包括使用 virt-manager 创建客户端的一般过程。第 3 章 中有安装各种通用操作系统的分解说明。

## 2.3. 使用 PXE 安装客户端

这部分论述了使用 PXE 安装客户端所需步骤。PXE 客户端安装需要共享网络设备，即我们所说的无论桥接。下面的过程包括桥接创建以及使用桥接进行 PXE 安装的步骤。

### 1. 创建新桥接

- a. 在 `/etc/sysconfig/network-scripts` 目录中创建新网络脚本。这个示例创建了名为 `ifcfg-installation` 的文件，它生成名为 *installation* 的桥接。

```
# cd /etc/sysconfig/network-scripts/  
# vim ifcfg-installation  
DEVICE=installation  
TYPE=Bridge  
BOOTPROTO=dhcp  
ONBOOT=yes
```

**Warning**

The line, `TYPE=Bridge`, is case-sensitive. It must have uppercase 'B' and lower case 'ridge'.

- b. Start the new bridge.

```
# ifup installation
```

- c. 还没有在新桥接中添加接口。请使用 `brctl show` 命令查看系统中网络桥接的详情。

```
# brctl show
bridge name  bridge id          STP enabled  interfaces
installation 8000.000000000000    no
virbr0       8000.000000000000    yes
```

virbr0 是 libvirt 在默认以太网设备中用于网络地址转换 (NAT) 的默认桥接。

2. 在新桥接中添加接口  
为该接口编辑配置文件。在该配置文件中添加 BRIDGE 参数，使用在前一步中创建的桥接名称。

```
# Intel Corporation Gigabit Network Connection
DEVICE=eth1
BRIDGE=installation
BOOTPROTO=dhcp
HWADDR=00:13:20:F7:6E:8E
ONBOOT=yes
```

编辑配置文件后，重新启动联网或者重新引导。

```
# service network restart
```

使用 `brctl show` 命令验证添加的接口：

```
# brctl show
bridge name  bridge id          STP enabled  interfaces
installation 8000.001320f76e8e    no           eth1
virbr0       8000.000000000000    yes
```

3. 安全性配置

Configure iptables to allow all traffic to be forwarded across the bridge.

```
# iptables -I FORWARD -m physdev --physdev-is-bridged -j ACCEPT
# service iptables save
# service iptables restart
```



### Disable iptables on bridges

Alternatively, prevent bridged traffic from being processed by iptables rules. In `/etc/sysctl.conf` append the following lines:

```
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
```

Reload the kernel parameters configured with `sysctl`

```
# sysctl -p /etc/sysctl.conf
```

4. 安装前重新启动 libvirt  
Restart the libvirt daemon.

```
# service libvirtd reload
```

桥接已配置，您现在可以开始安装了。

### 使用 `virt-install` 进行 PXE 安装

对于 `virt-install` 安装，请附加 `--network=bridge:installation` 安装参数，其中 *installation* 是您桥接的名称。对于 PXE 安装请使用 `--pxe` 参数。

```
# virt-install --accelerate --hvm --connect qemu:///system \
  --network=bridge:installation --pxe \
  --name EL10 --ram=756 \
  --vcpus=4
  --os-type=linux --os-variant=rhel5
  --file=/var/lib/libvirt/images/EL10.img \
```

### 例 2.3. 使用 `virt-install` 进行 PXE 安装

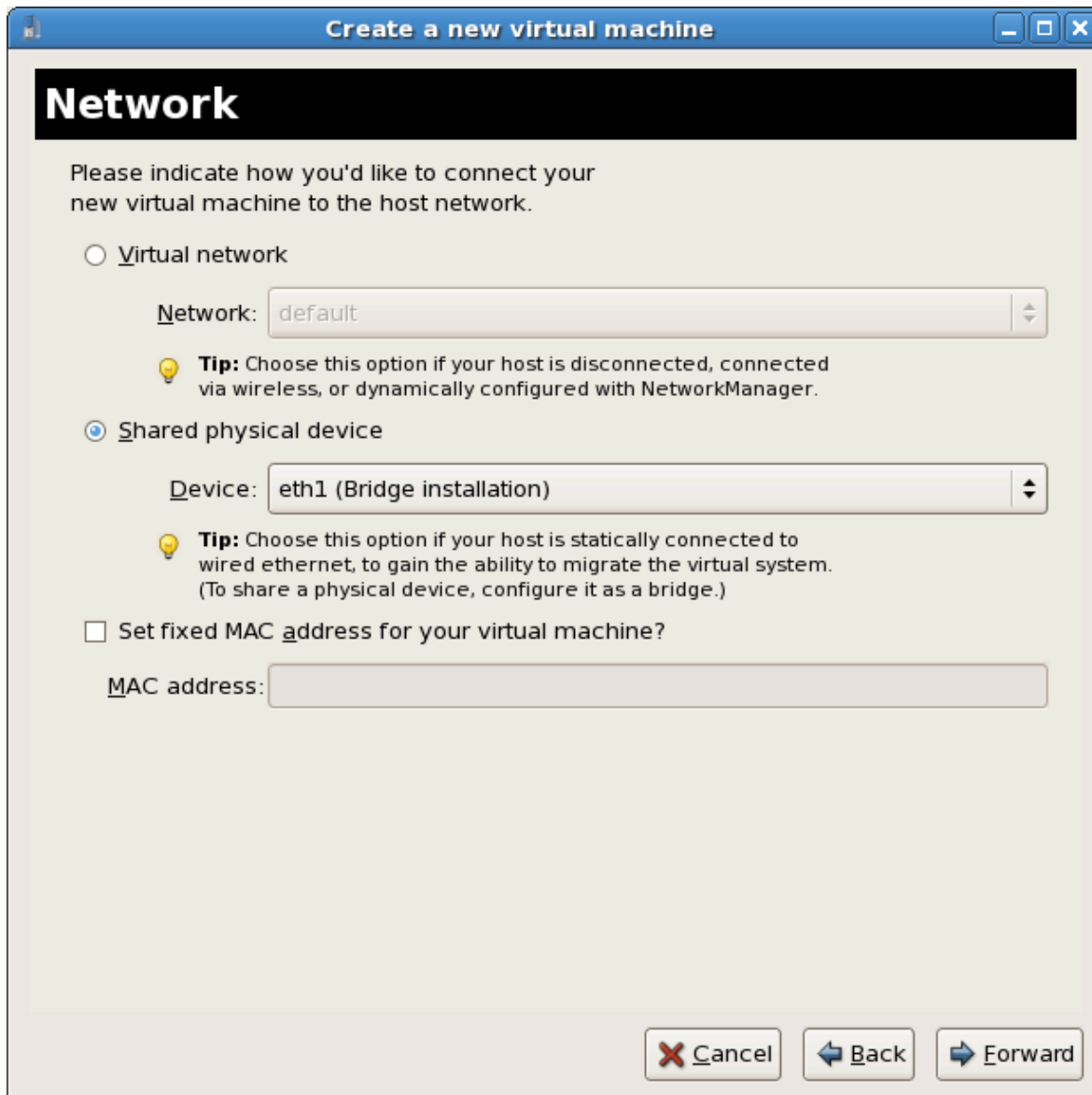
### 使用 `virt-manager` 进行 PXE 安装

红帽企业版 Linux 的每个具体版本都有详细的安装说明，其它 Linux 发行本、Solaris 和 Windows 的说明请参考 [3](#)。

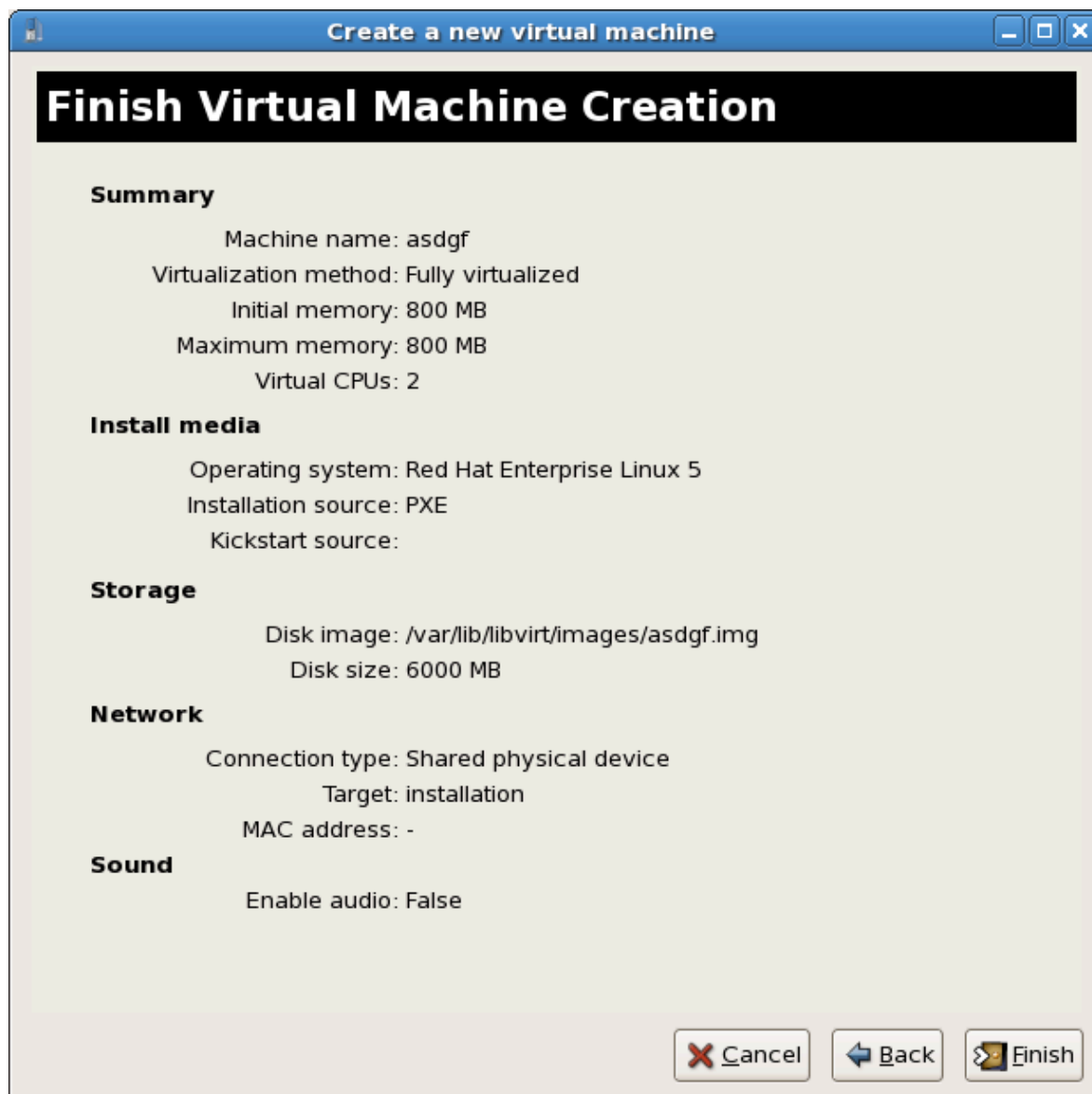
1. 选择 PXE  
选择 PXE 作为安装方法。



2. 选择桥接  
选择「共享物理设备」并选择在前面的步骤中创建的桥接。



- 3. 开始安装  
准备开始安装。



发送 DHCP 请求，找到有效 PXE 服务器后即可开始安装过程。





# 客户端操作系统安装过程

本章论述了如何在红帽企业版 Linux 的虚拟环境中安装各种客户端操作系统。要了解基本过程，请参考 [2.0 虚拟化](#)。

## 3.1. 将红帽企业版 Linux 5 作为半虚拟客户端安装

这部分论述了如何安装红帽企业版 Linux 5 作为半虚拟客户端。半虚拟化要比全虚拟化速度更快，并支持所有全虚拟化的优点。半虚拟化需要一个特殊支持的内核，即 `kernel-xen` 内核。



### 半虚拟化的重要注释

半虚拟化只适用于 Xen 管理程序。半虚拟化不可用于 KVM 管理程序。

确定您在开始安装前可进行根访问。

这个方法可从远程服务器安装红帽企业版 Linux。此处的安装说明与使用实时 CD-ROM 进行最小安装类似。

使用 `virt-manager` 或者 `virt-install` 创建半虚拟红帽企业版 Linux 5 客户端。有关 `virt-manager` 的步骤请参考 [2.2 “使用 virt-manager”](#)。

使用基于 `virt-install` 工具的命令行创建半虚拟客户端。`--vnc` 选项显示图形安装。这个示例中的客户端名称为 `rhe15PV`，磁盘映像文件为 `rhe15PV.dsk`，红帽企业版 Linux 5 安装树的本地镜像为 `ftp://10.1.1.1/trees/RHEL5-B2-Server-i386/`。使用您系统和网络的准确值替换这些值。

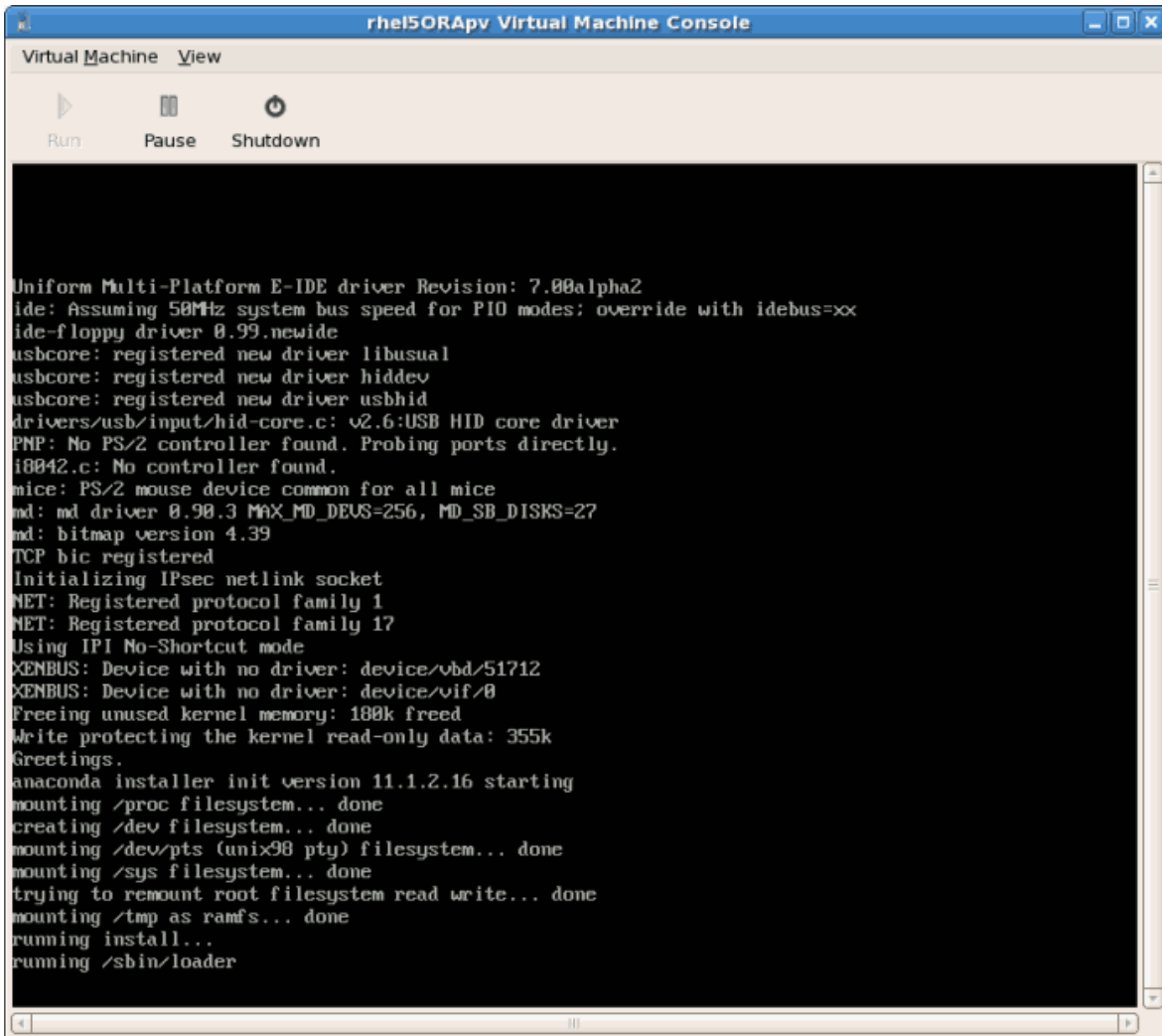
```
# virt-install -n rhe15PV -r 500 \  
-f /var/lib/libvirt/images/rhe15PV.dsk -s 3 --vnc -p \  
-l ftp://10.1.1.1/trees/CentOS5-B2-Server-i386/
```



### 自动安装

可不使用图形界面或者手动输入安装红帽企业版 Linux。使用 Kickstart 文件自动完成安装进程。

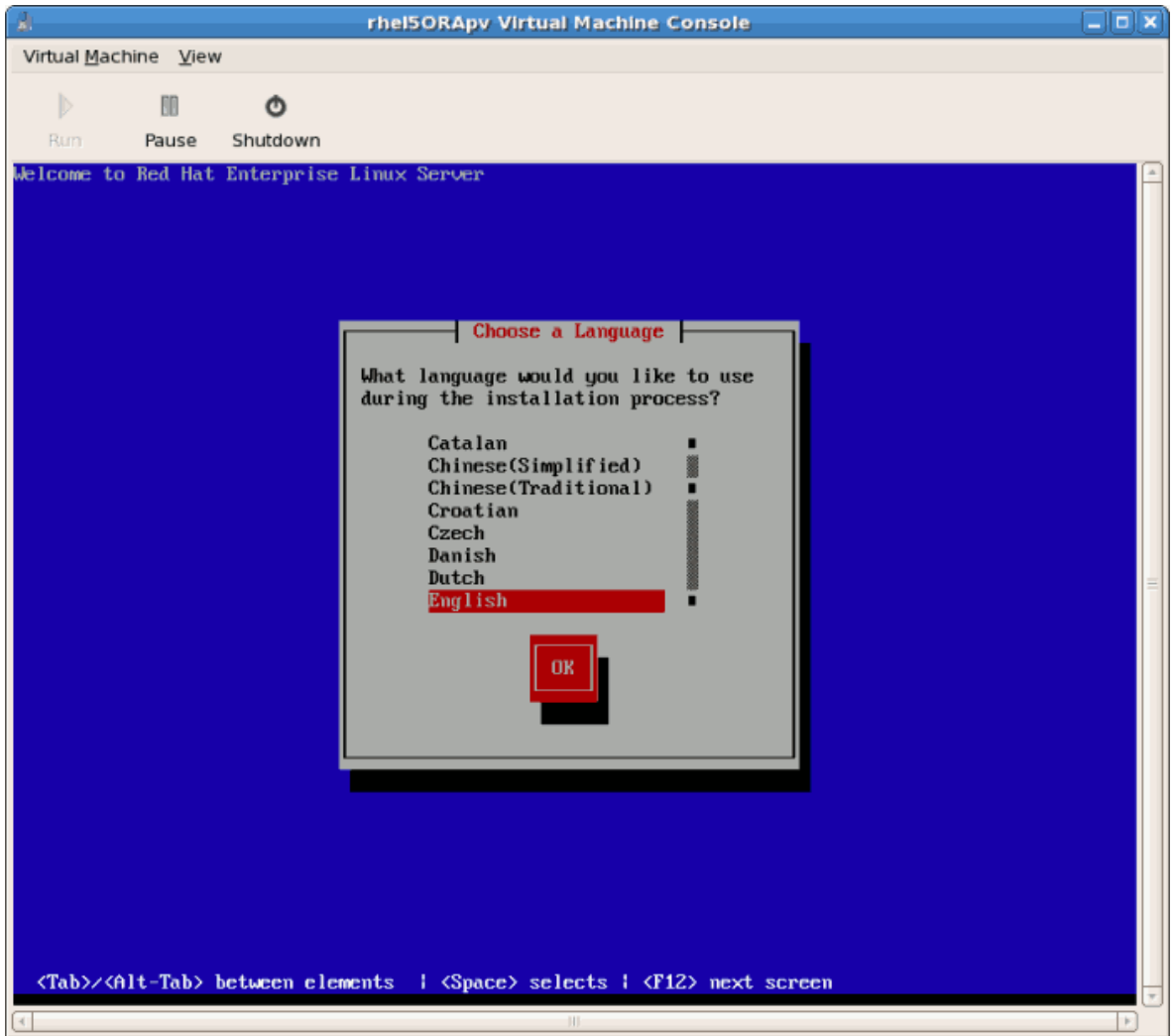
使用两种方法之一打开这个窗口，显示您客户端的初始化引导阶段：



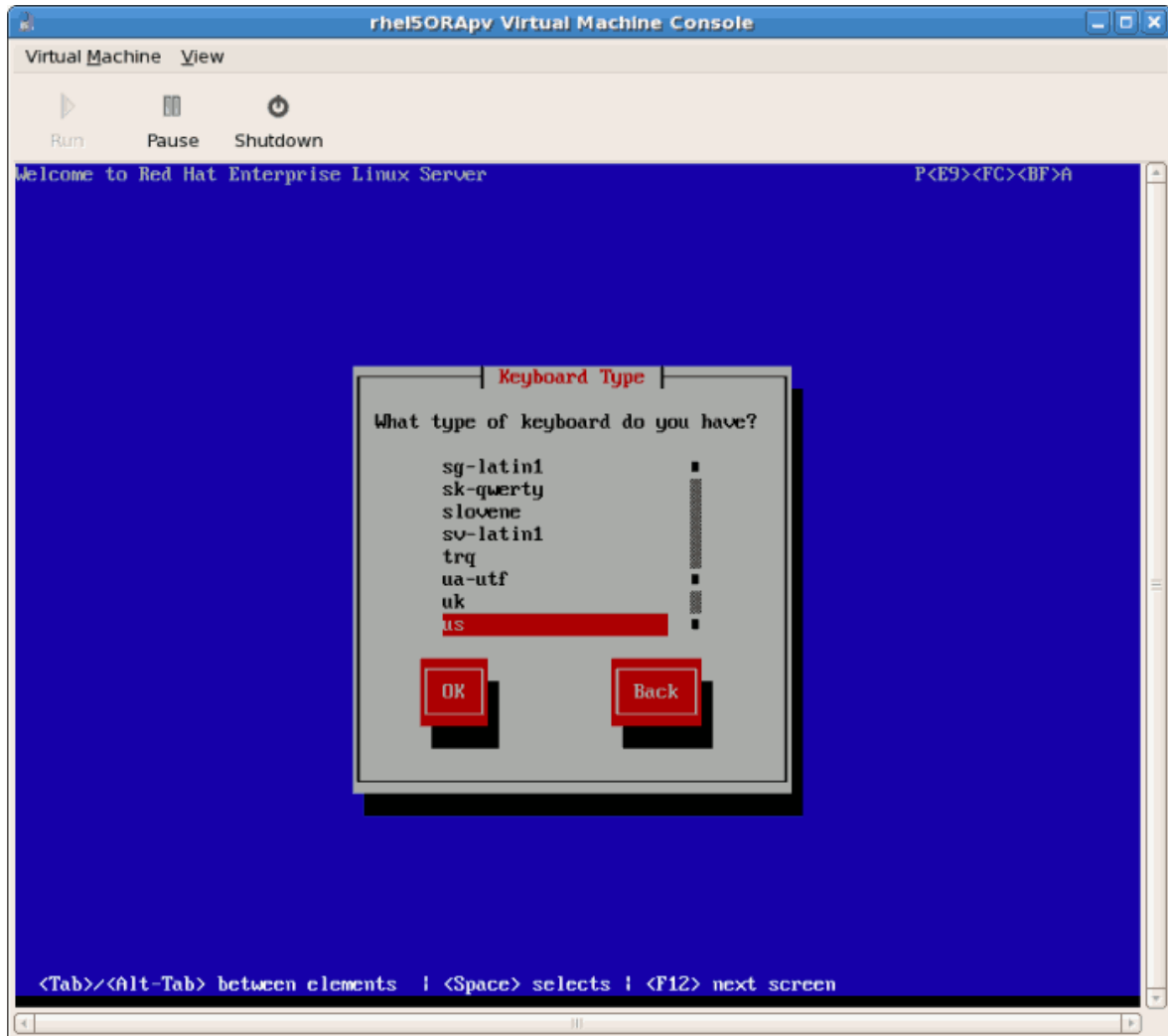
当您的客户端完成初始化引导后将开始红帽企业版 Linux 的标准安装过程。大多数系统的默认回答是接受。

### 过程03.1.0半虚拟红帽企业版 Linux 客户端安装过程

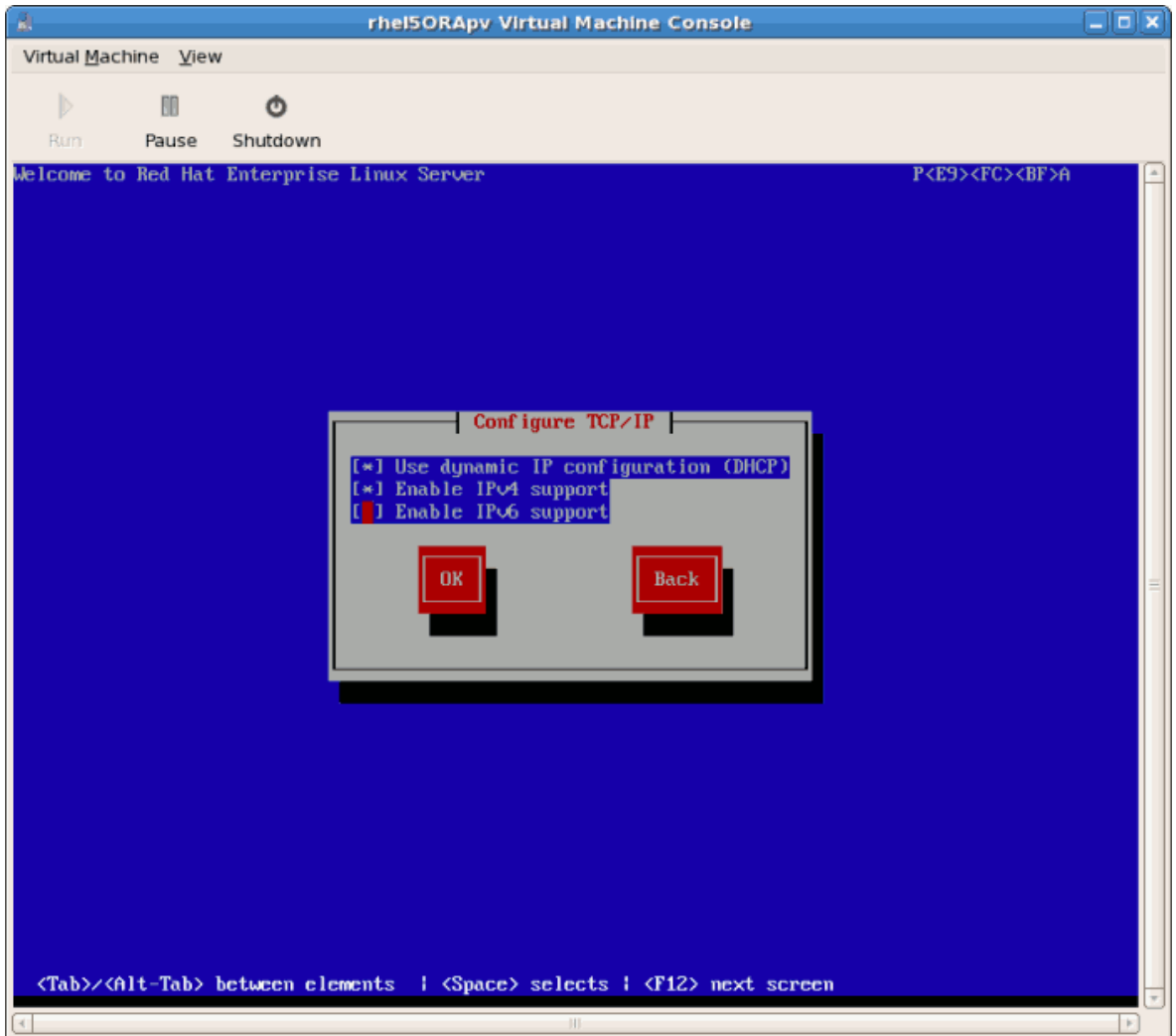
1. 选择语言并点击 确定。



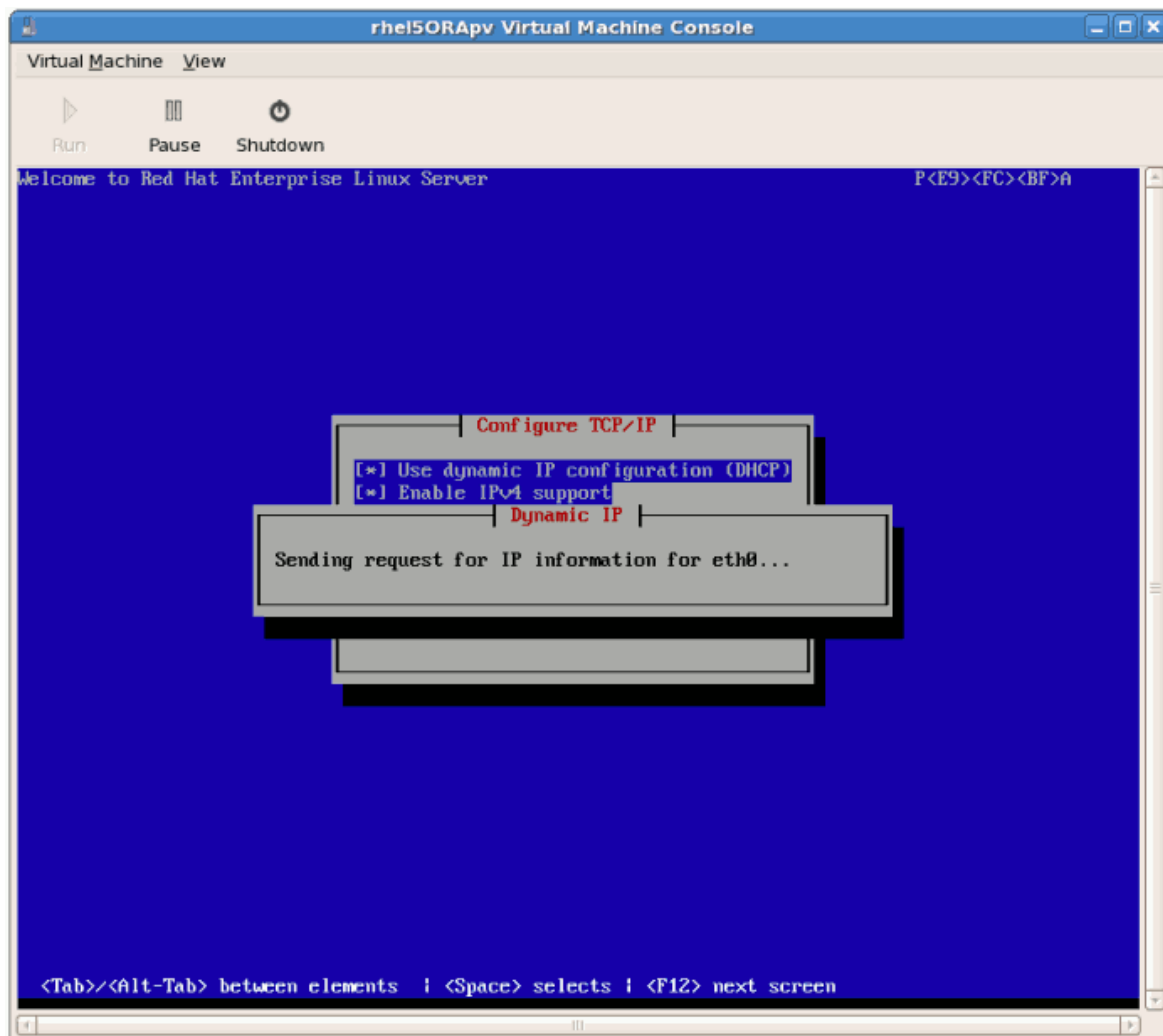
2. 选择键盘布局并点击 确定。



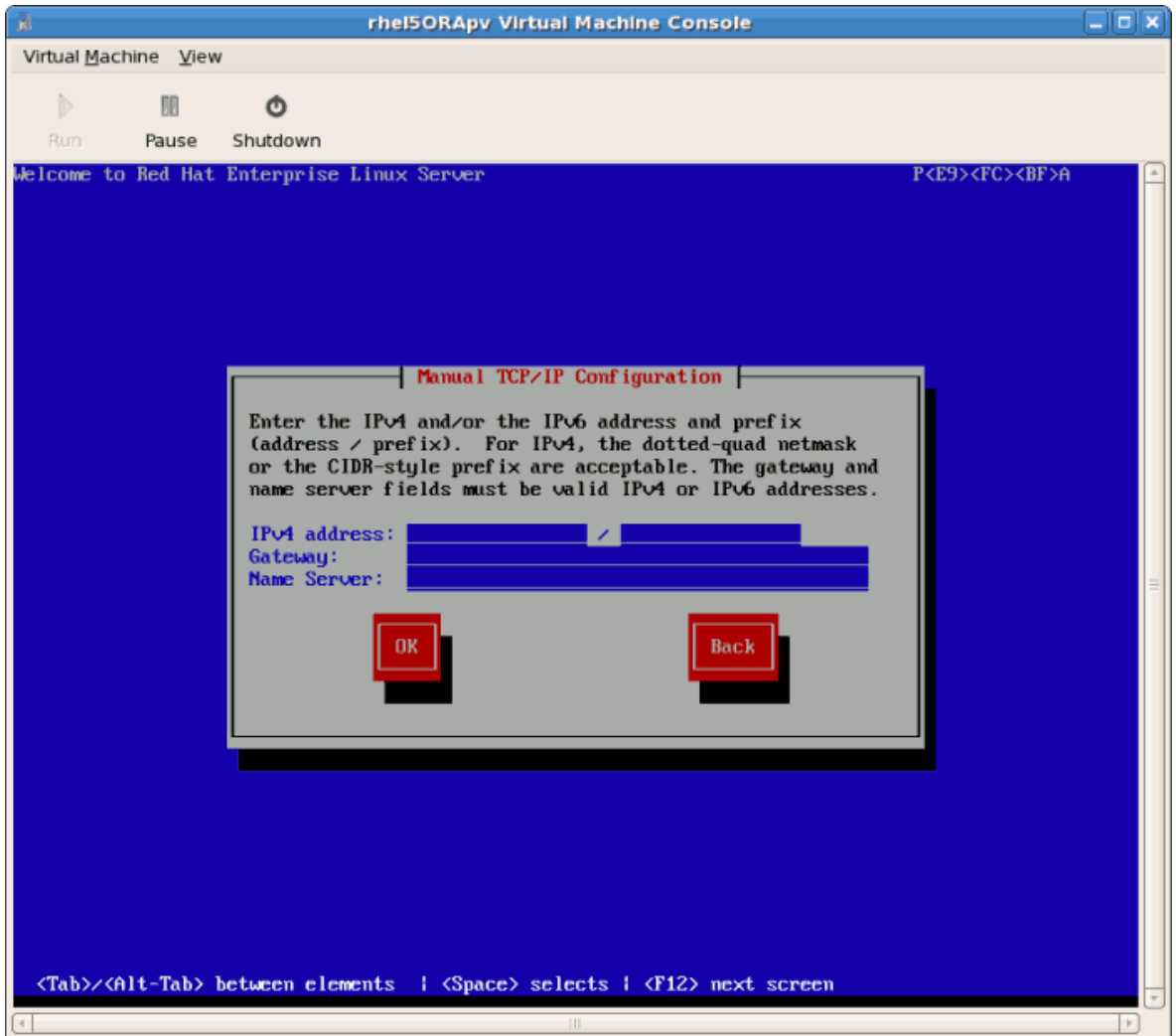
3. 分配客户端网络地址。您可以使用 DHCP（如下所示），也可以使用静态地址：



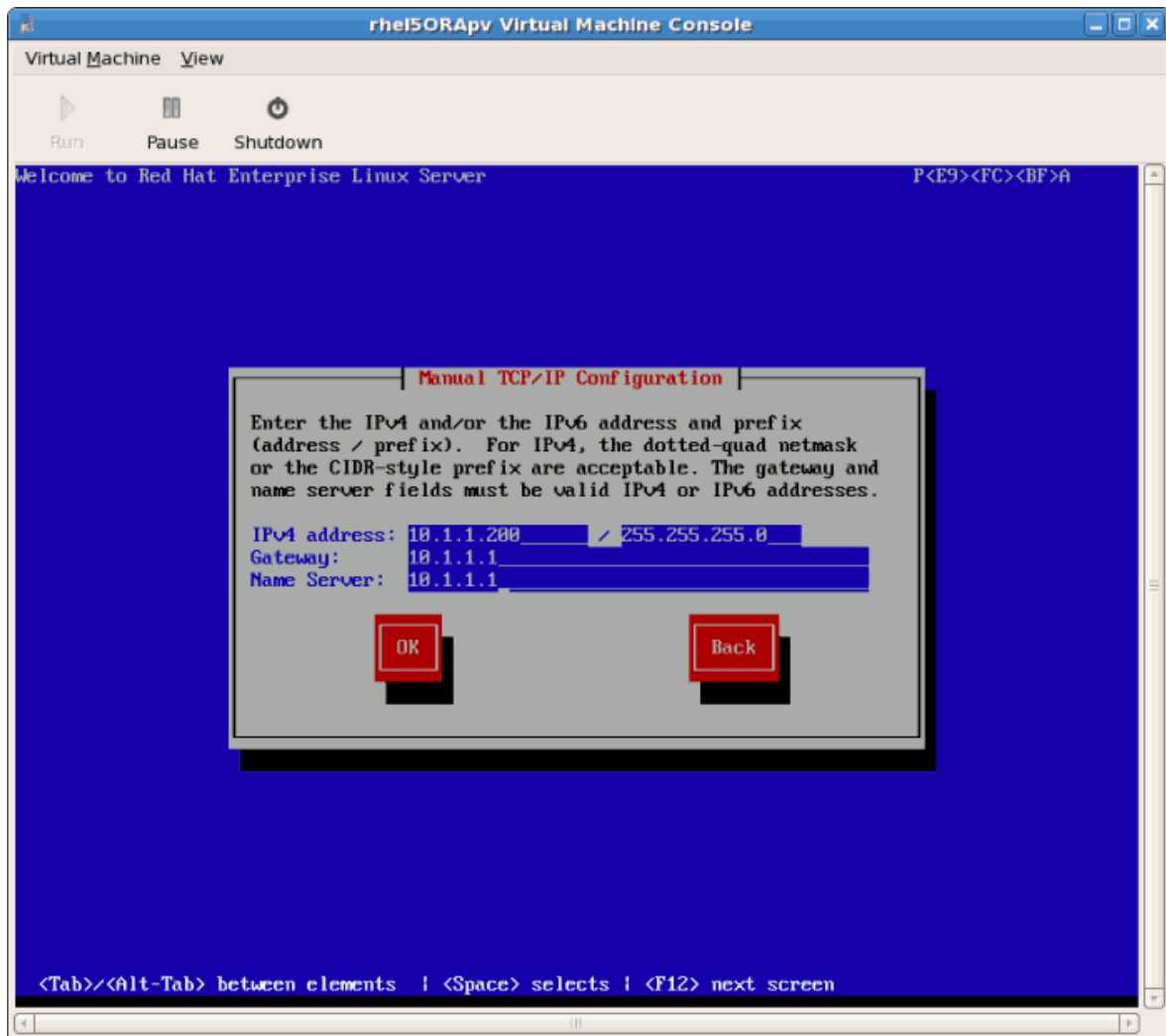
4. 如果您选择 DHCP，则安装程序现在会试图获取 IP 地址：



5. 如果您为客户端选择静态 IP 地址，则会出现以下提示。请输入客户端联网配置详情：
- a. 请输入有效 IP 地址，确定您输入的 IP 地址可到达有安装树的服务器。
  - b. 输入有效子网掩码、默认网关以及名称服务器地址。
- 选择语言并点击 确定。

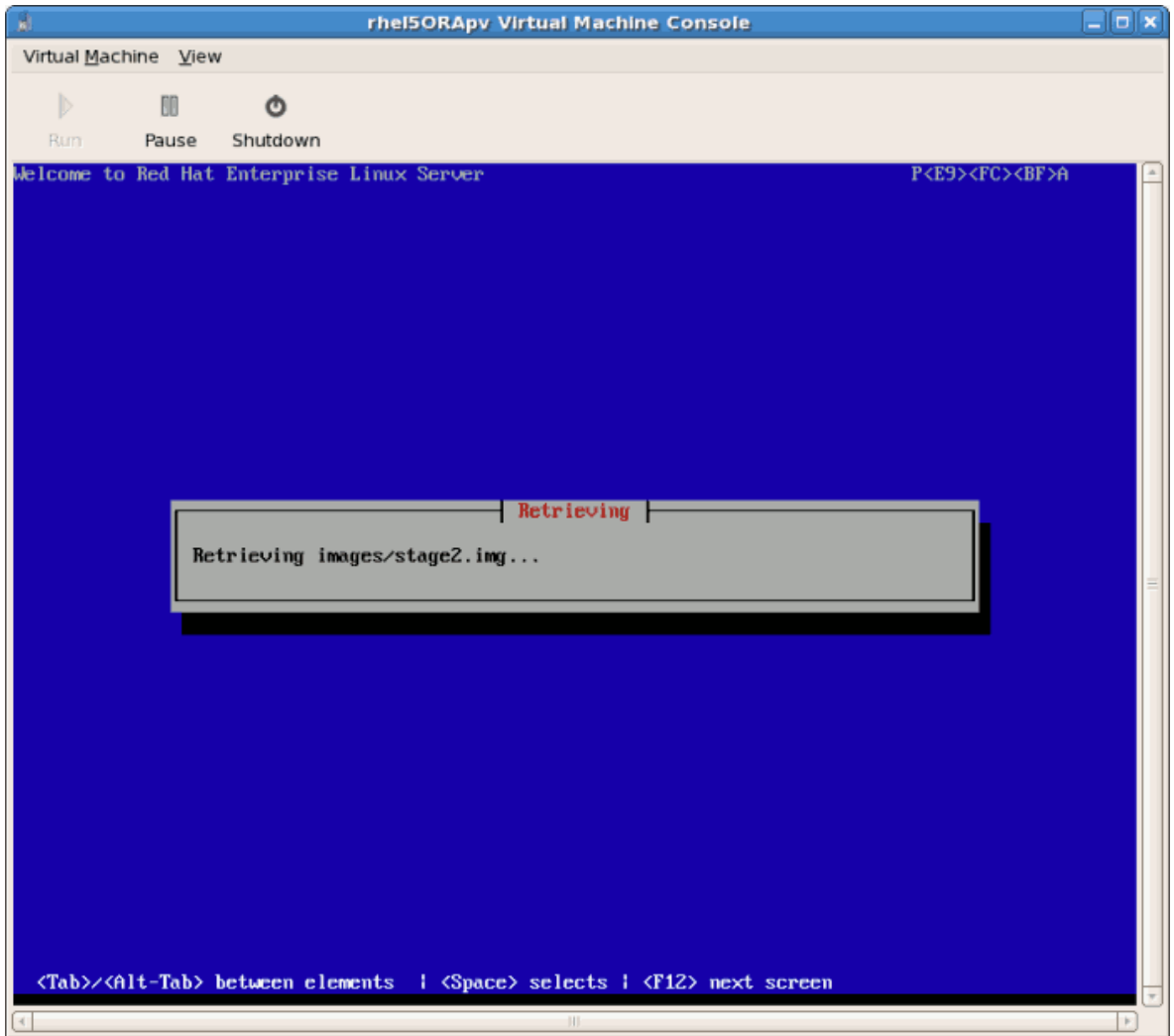


6. 这是一个静态 IP 地址配置示例：

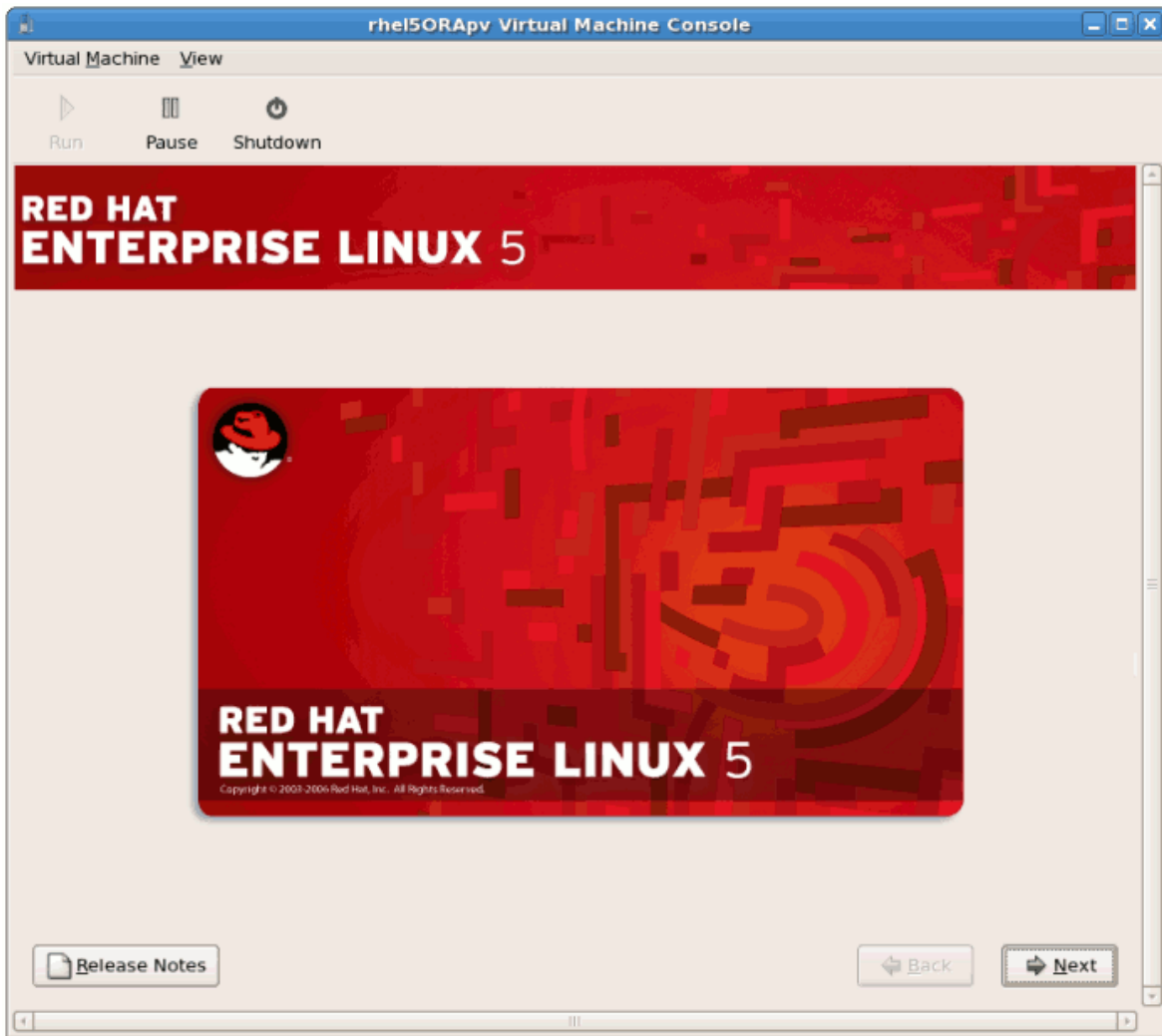




7. 安装进程现在从服务器中搜索所需文件：

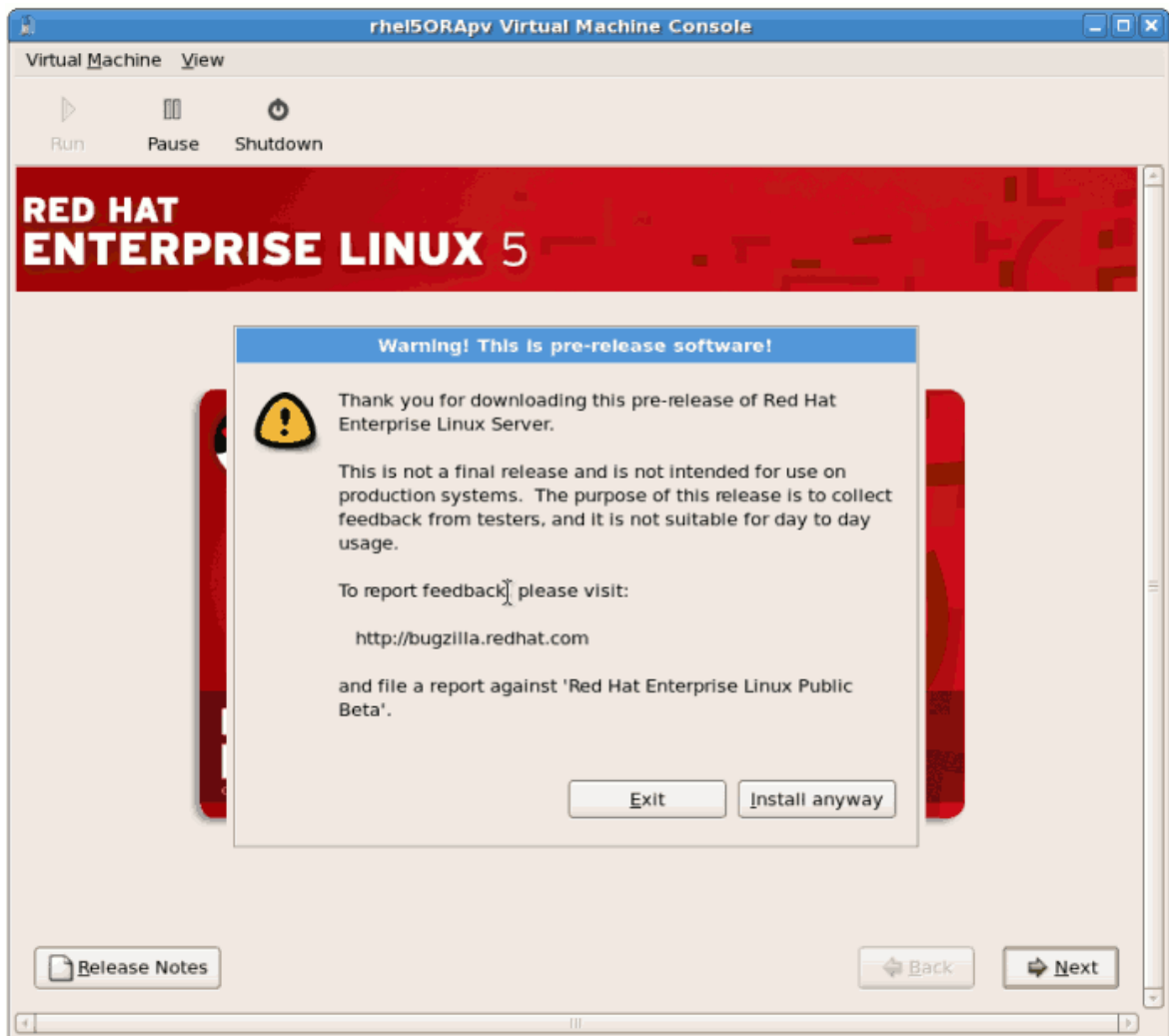




初始化步骤完成后启动图形安装进程。

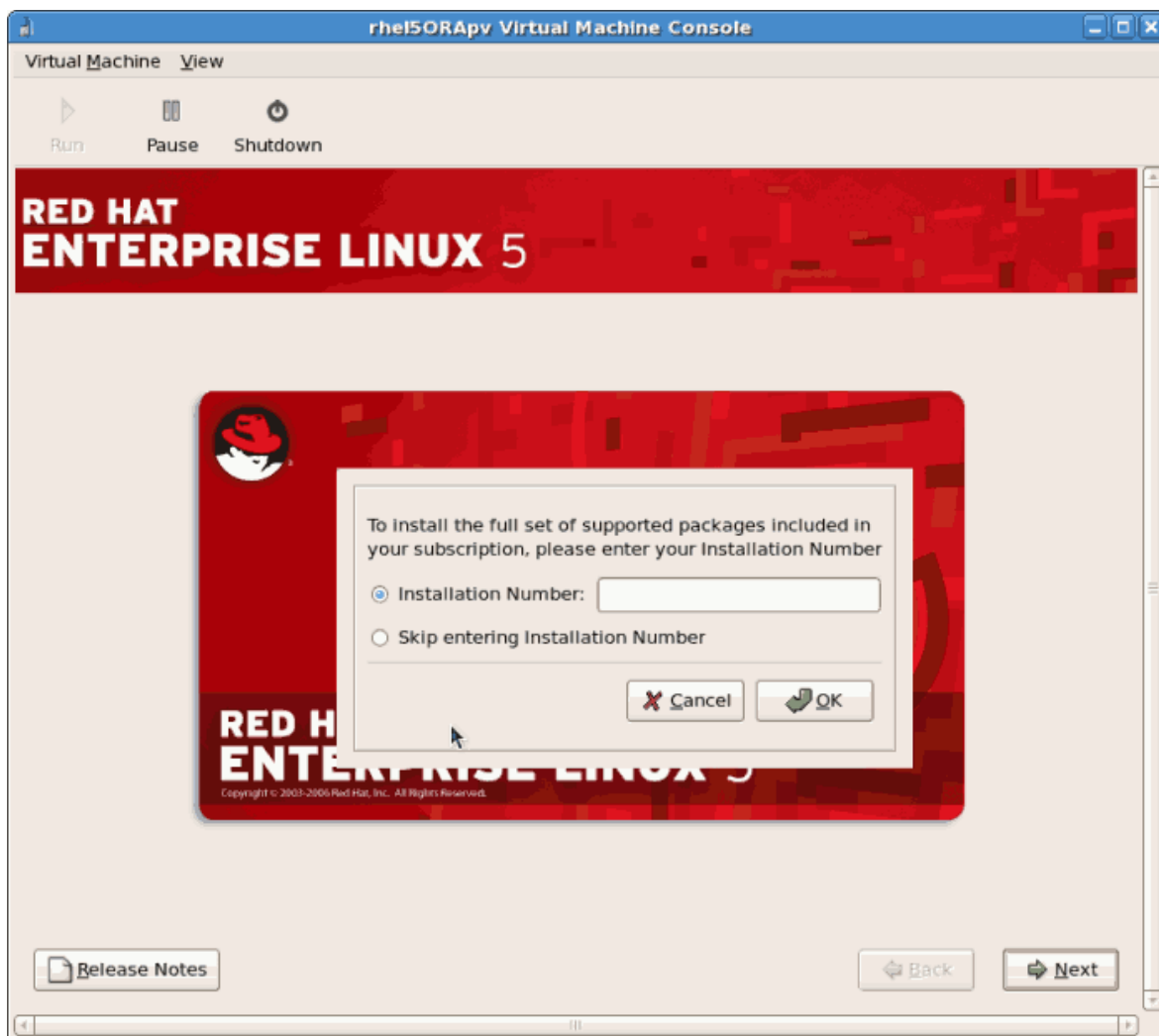



如果您正在安装  或者更早的发行版本，请确定您要安装该操作系统。点击  无论如何都安装，然后点击确定。



### 过程03.2.0图形安装过程

1. 输入有效注册码。如果您有一个有效 RHN 订阅密钥，请在 Installation Number 字段输入：

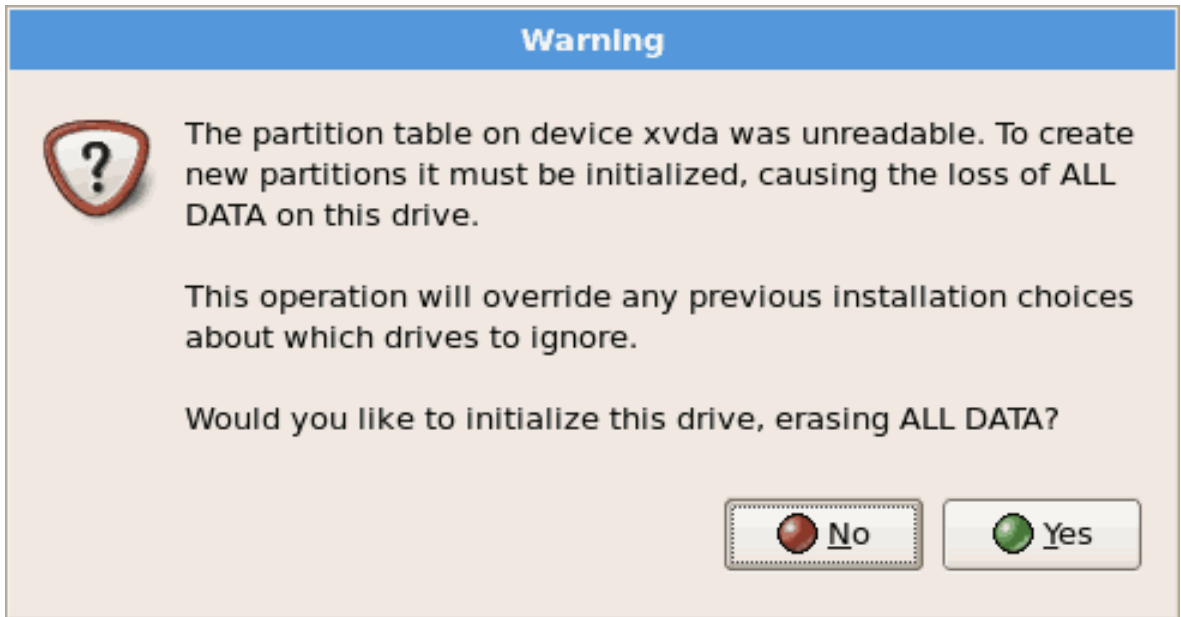


 Note

如果您跳过注册步骤，那么您可以在安装后使用 `rhn_register` 命令确认您的 fedora 网络帐户详情。`rhn_register` 命令需要根访问。

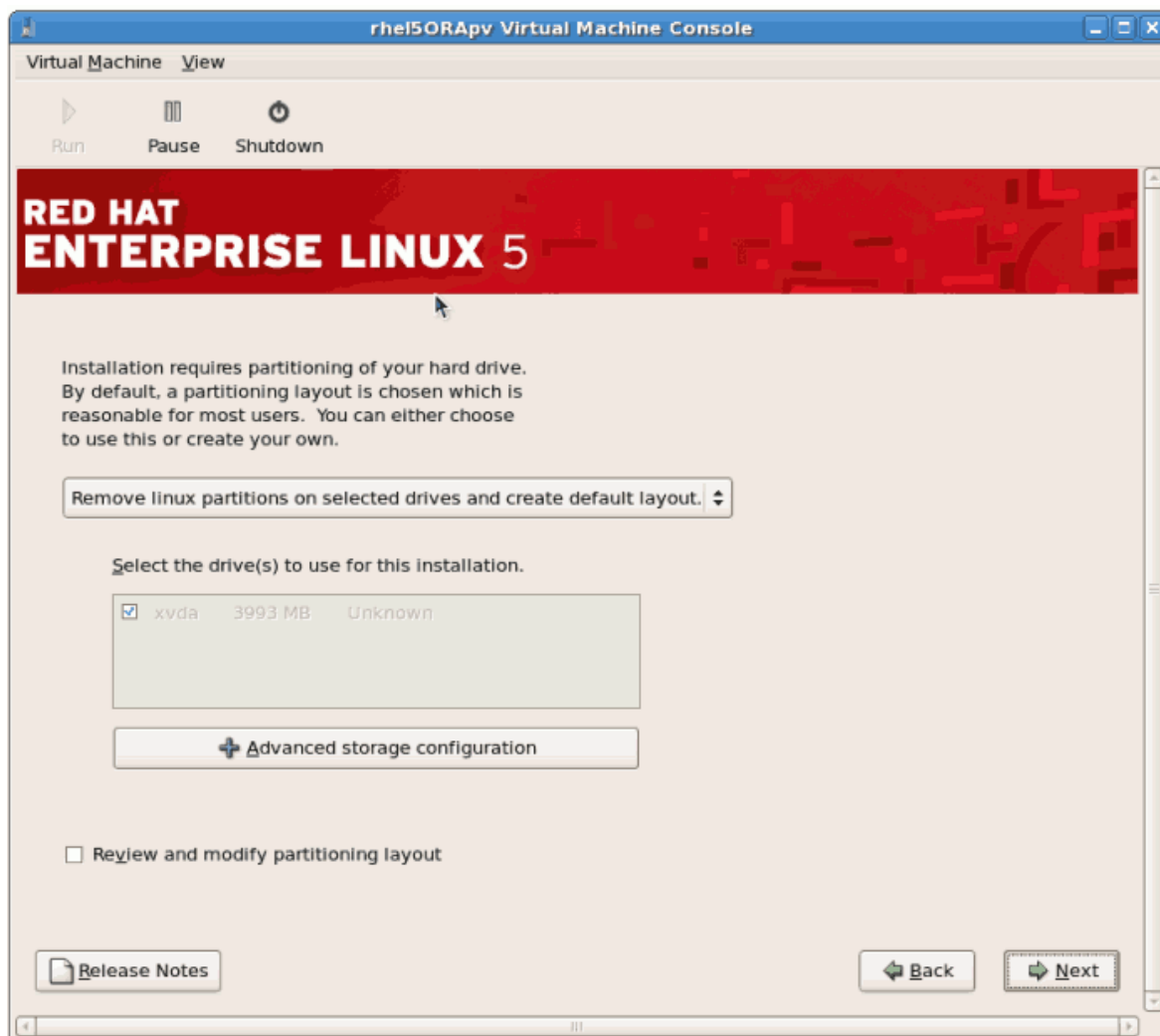
```
# rhn_register
```

2. 安装程序提示您确定想要在安装过程中删除您所选存储设备中的所有数据：



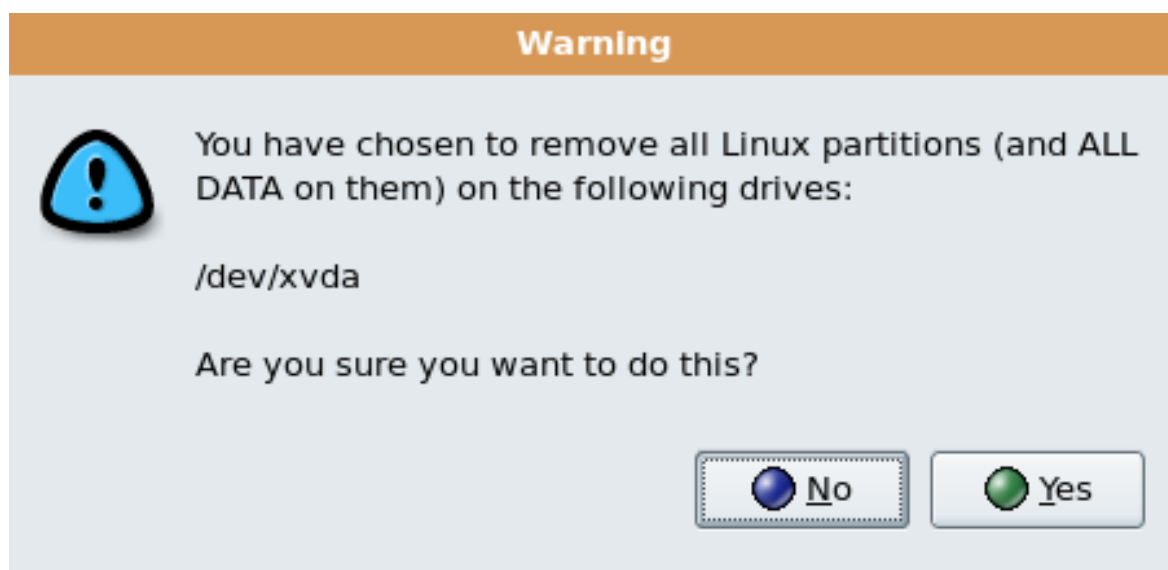
点击 是 继续。

3. 查看存储配置和分区布局。如果您要在客户端存储中使用 iSCSI，则可以选择高级存储配置。



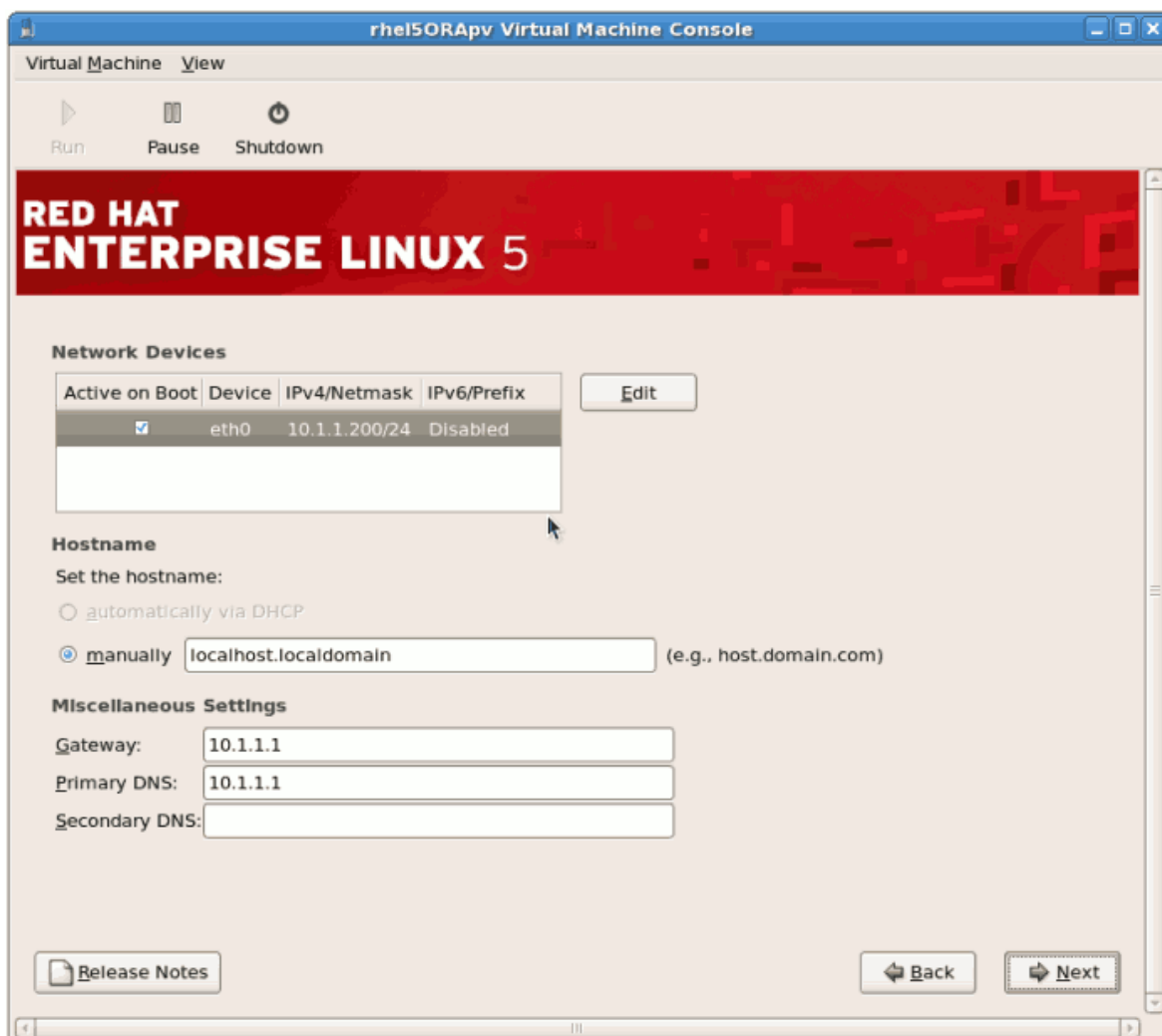
选择然后点击 下一步。

4. 为安装确定选择的存储。



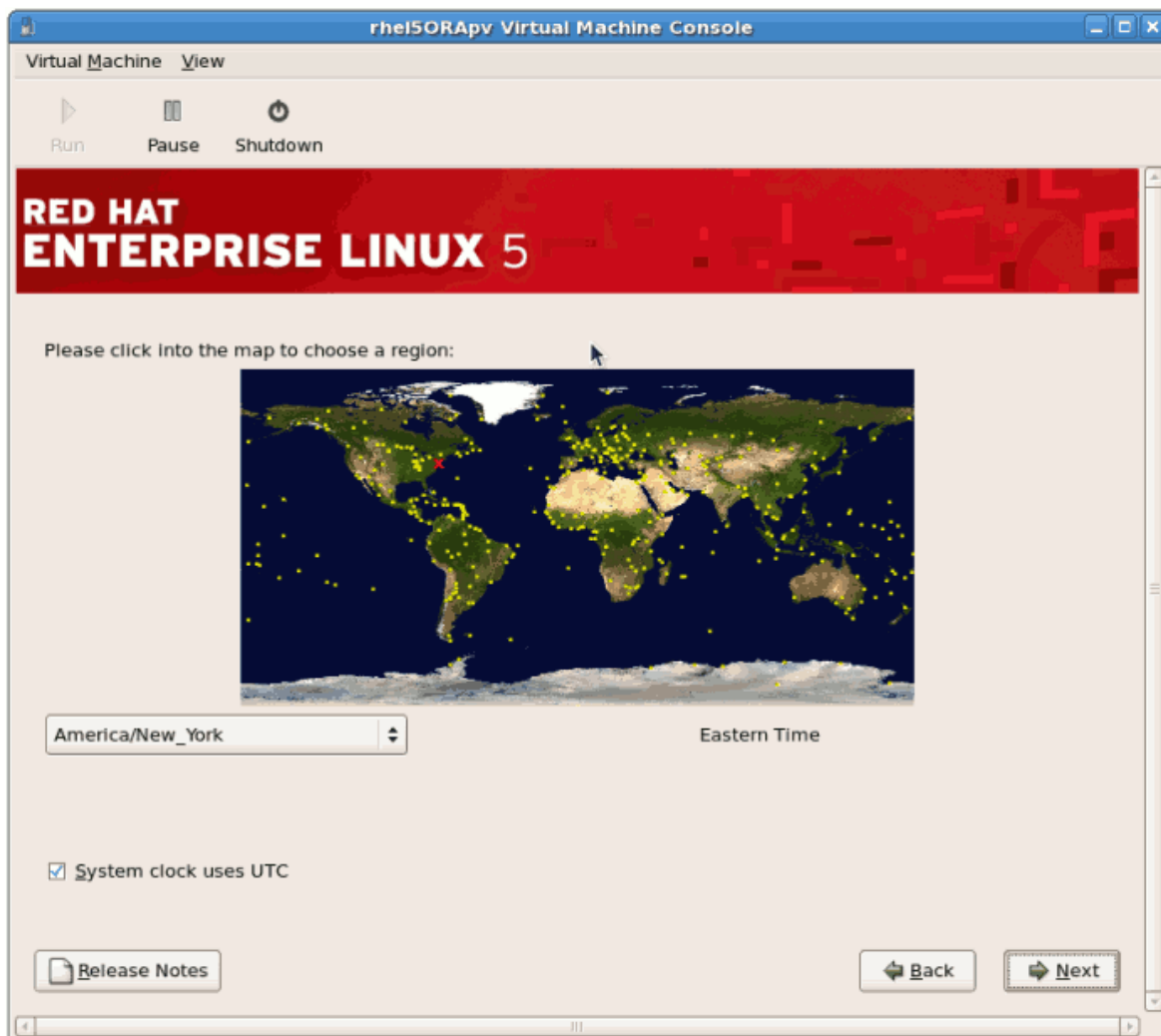
点击 是 继续。

5. 配置联网和主机名设定。这些设定使用早先安装过程输入的数据，需要时可修改这些设定。



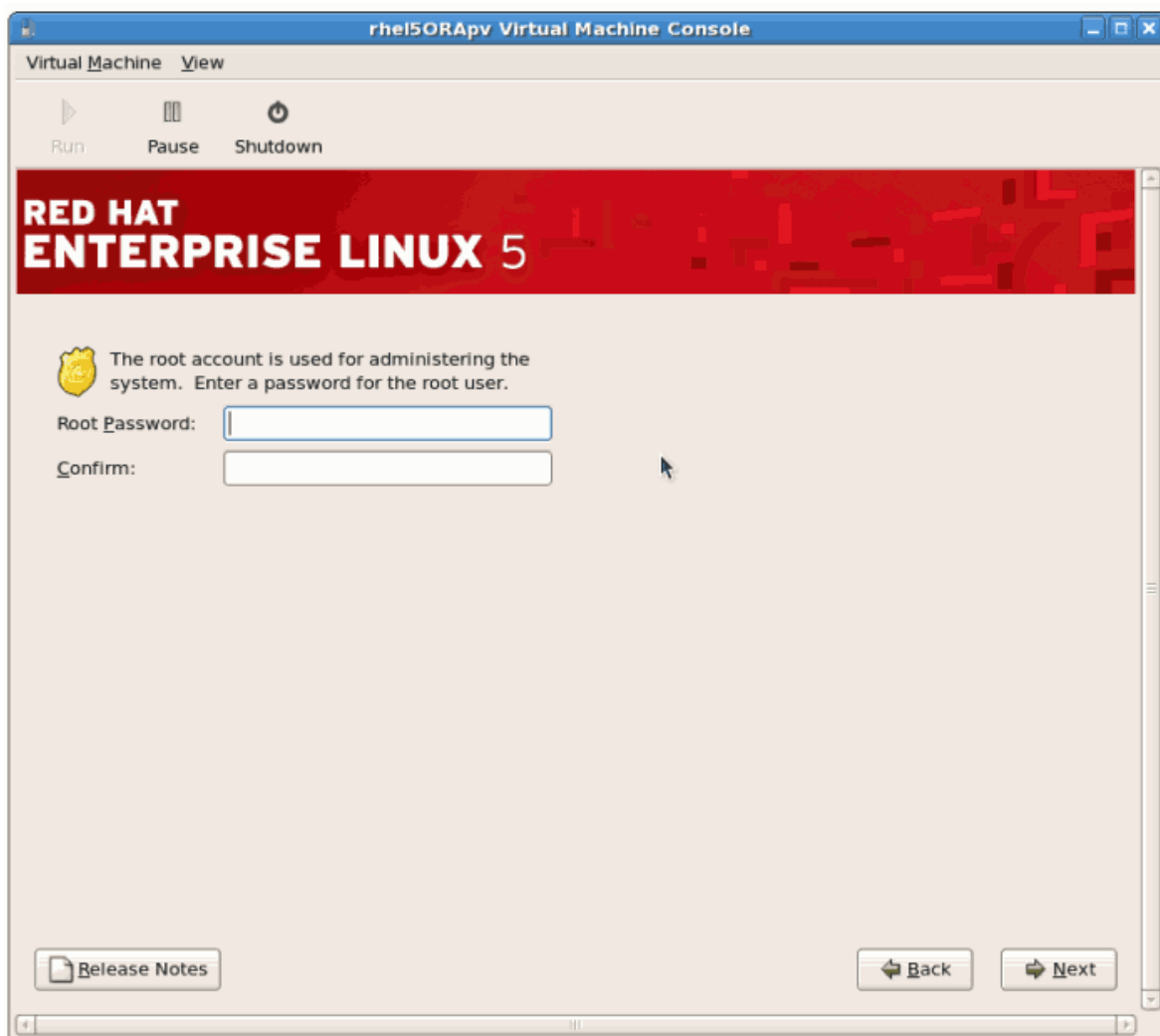
点击 确定 继续。

6. 为您的环境选择适当的时区。



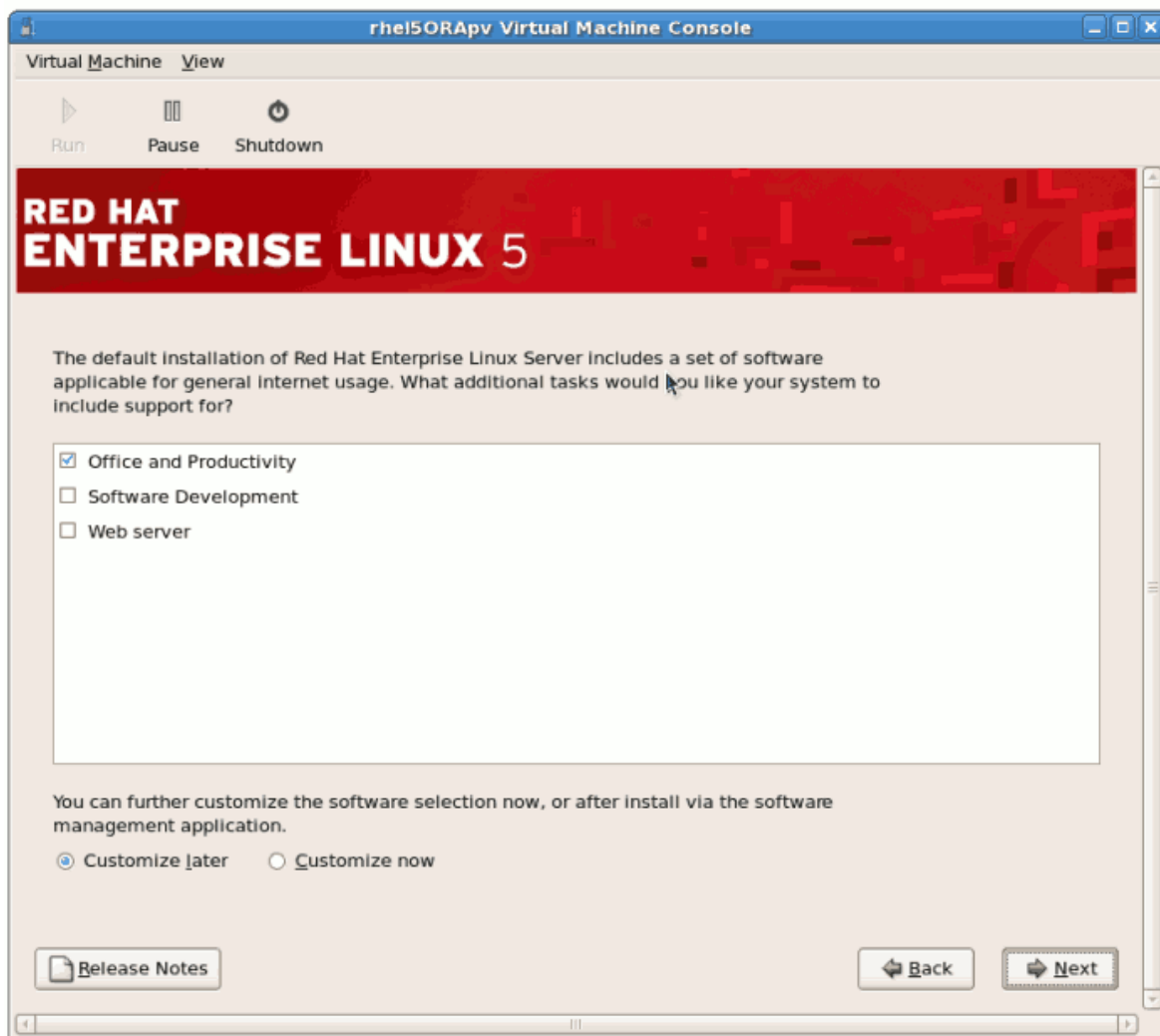
7. 为您的客户端输入根密码。





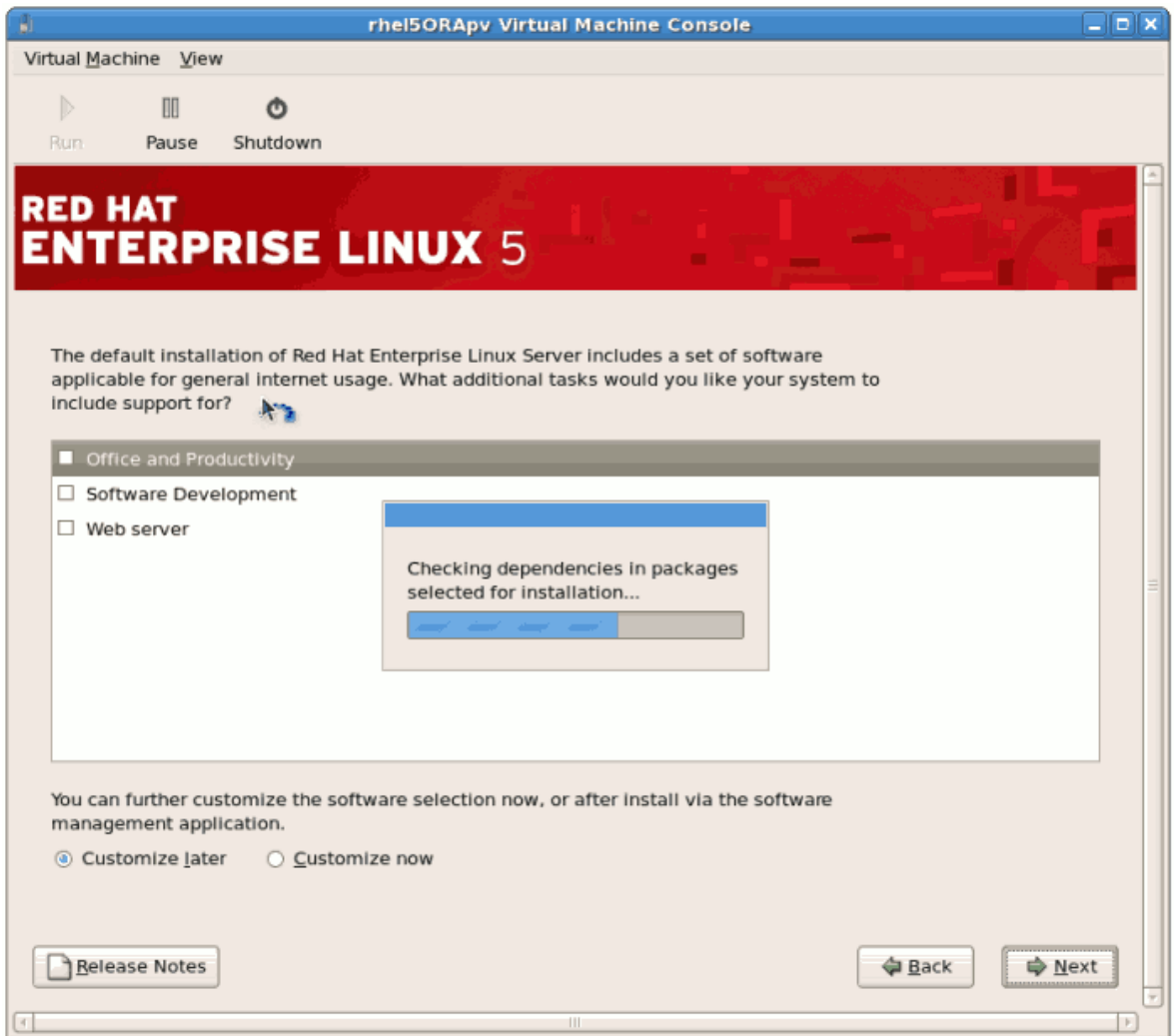
点击 下一步 继续。

8. 选择要安装的软件软件包。选择 现在定制 按钮。您必须在「System」目录中安装「kernel-xen」。半虚拟化需要「kernel-xen」软件包。

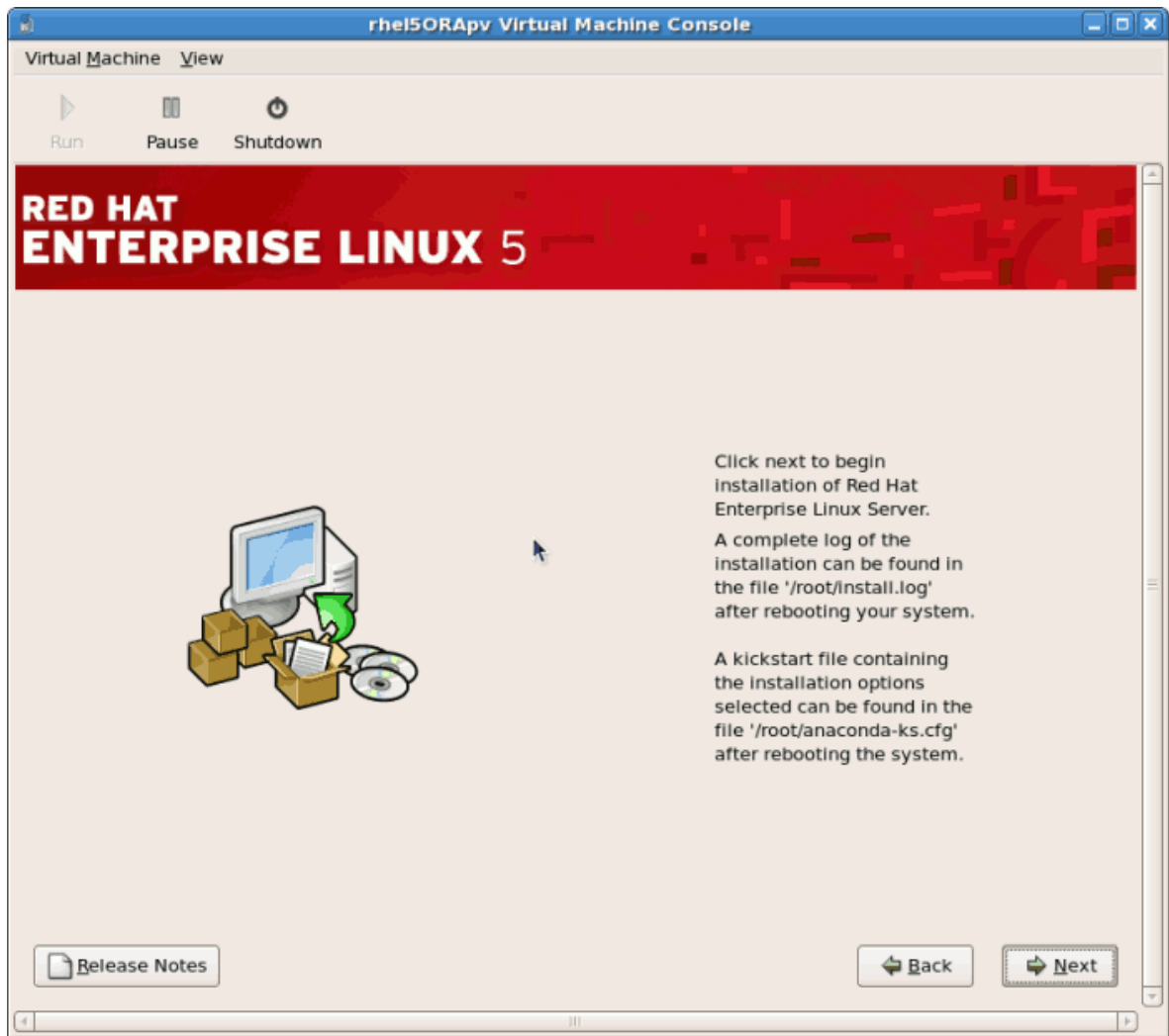


点击 下一步。

9. 计算相依性和空间要求。



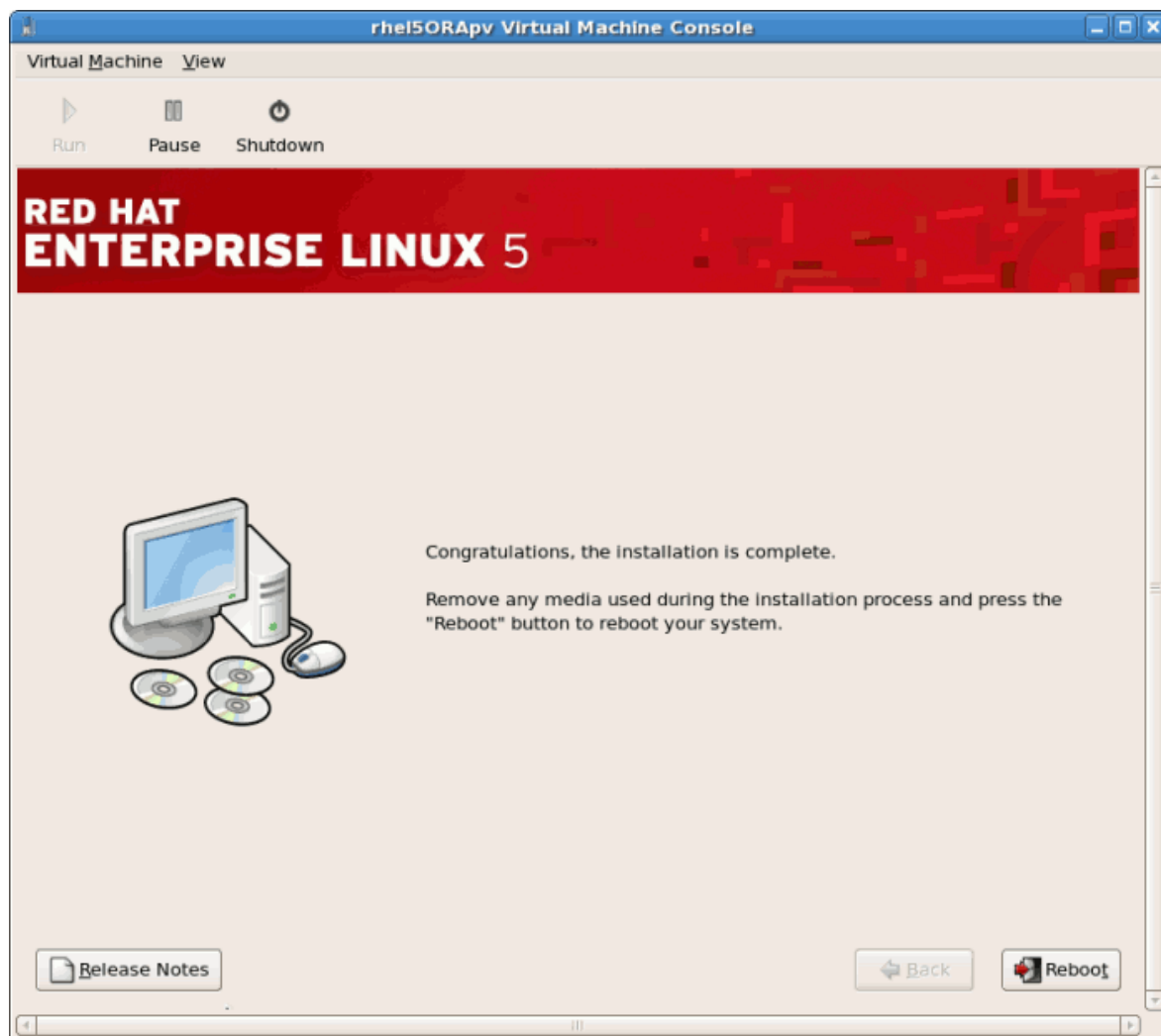
10. 在您确定安装相依性以及空间要求后，点击 下一步 按钮开始实际安装。



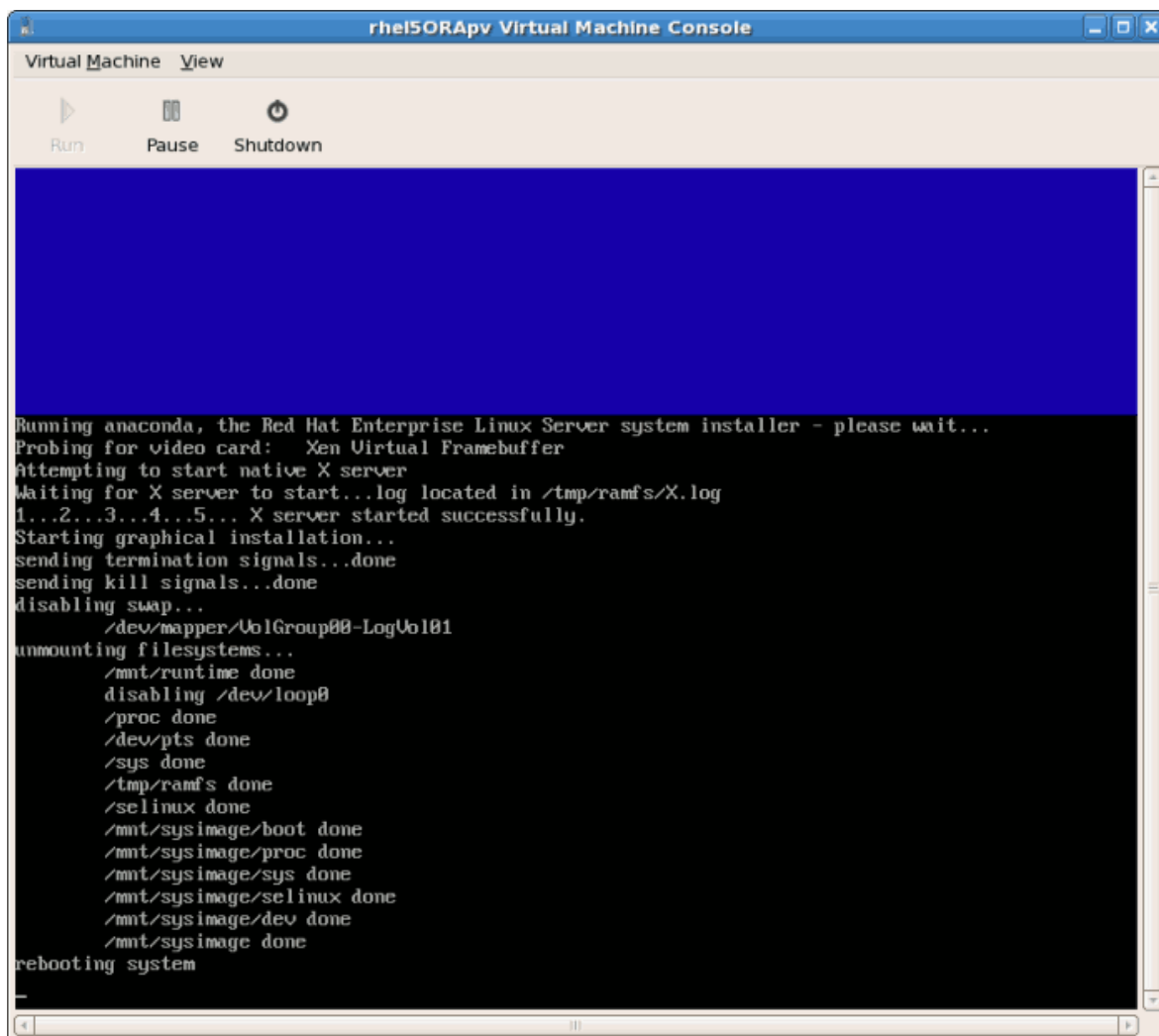
11. 自动安装所有选择的软件软件包。



12. 完成安装后重启您的客户端：



13. 客户端不会重启，而是会关闭.....



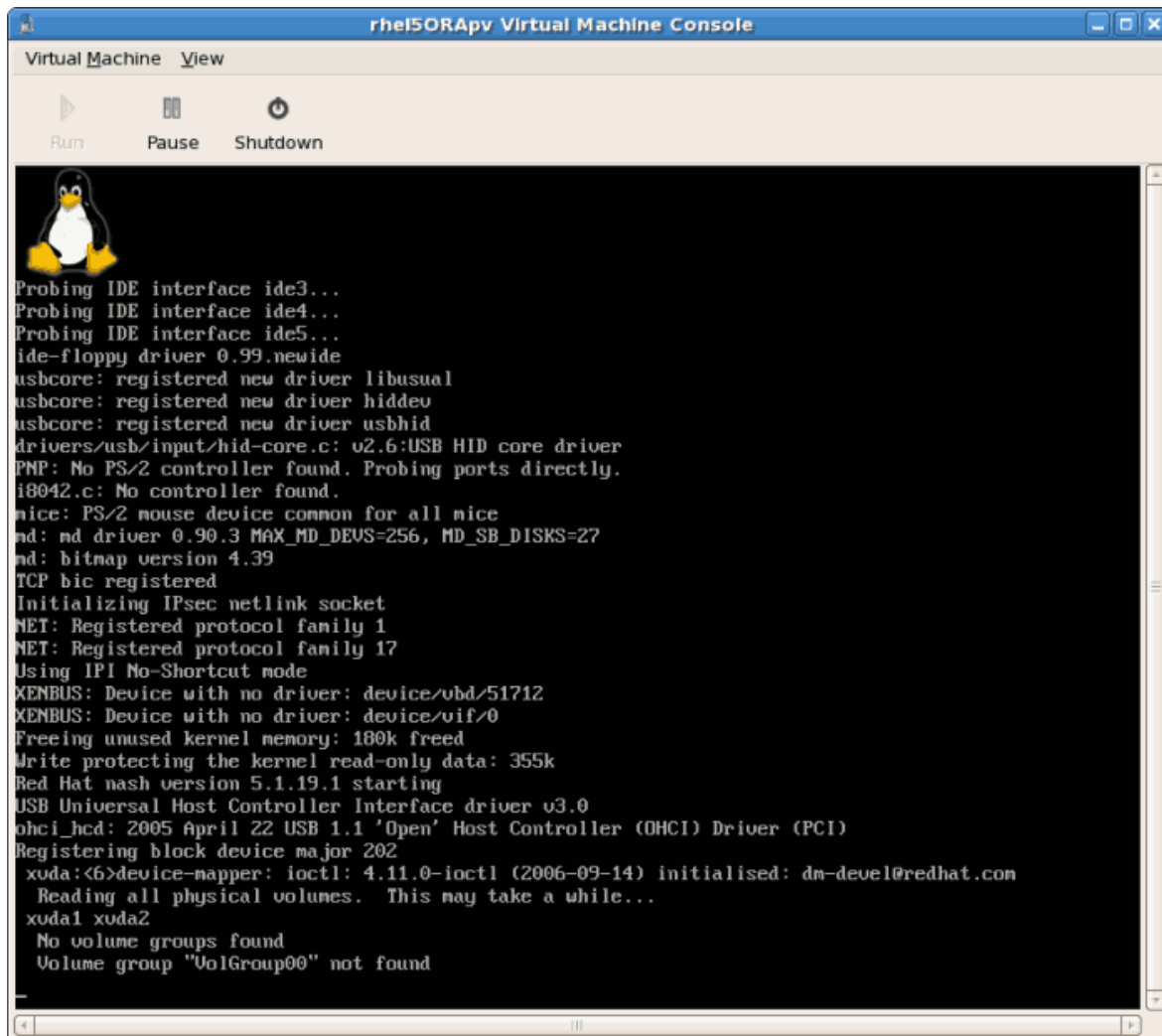
14. 引导该客户端。当您使用 [3.1](#) 中的 `virt-install` 时会选择客户端名称。如果您使用默认示例，则该名称为 `rhe15PV`。

运行：

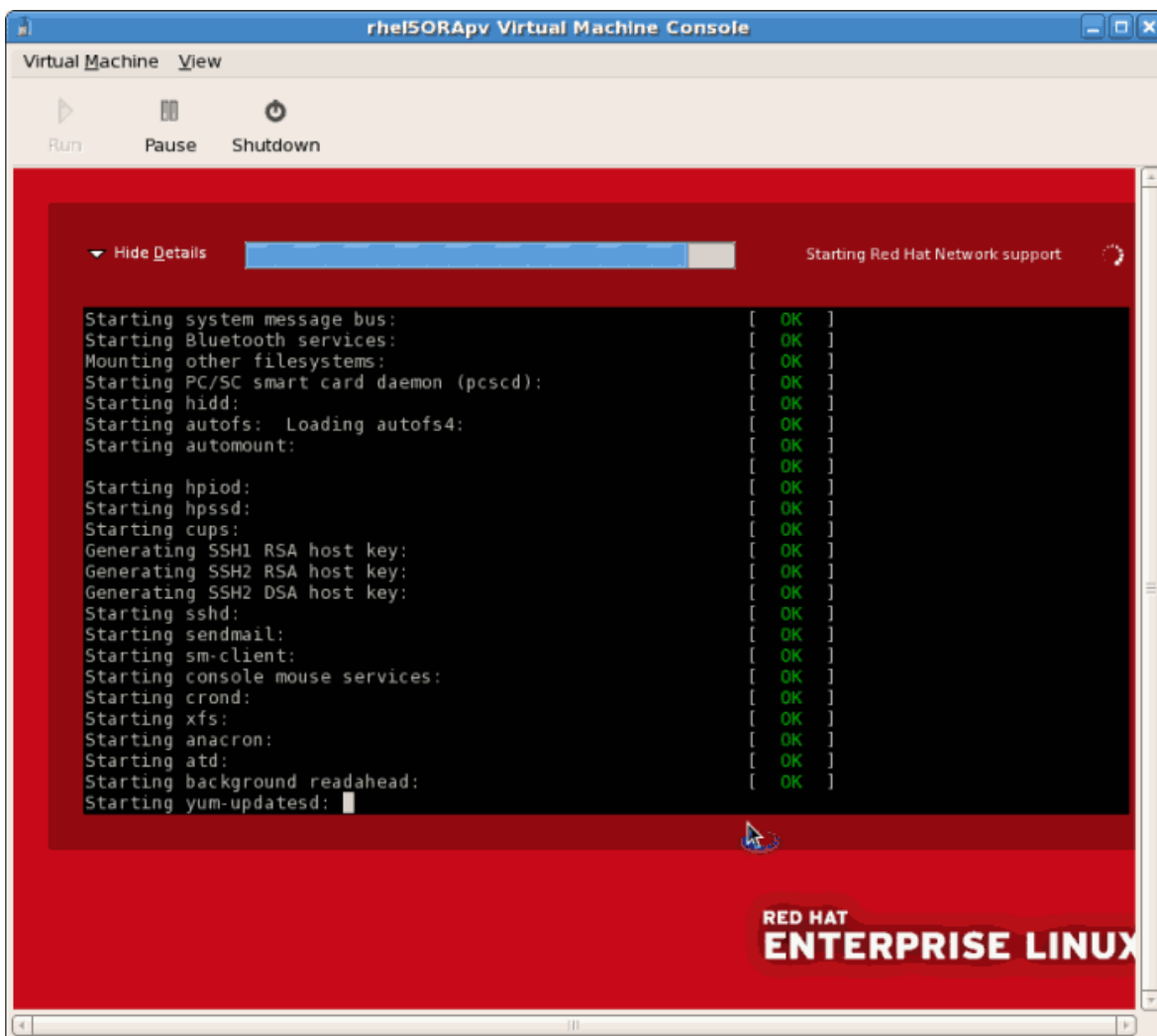
```
virsh reboot rhe15PV
```

另外，打开 `virt-manager`，选择您客户端的名称，点击 打开，然后点击 运行。

现在打开了显示客户端引导过程的 VNC 窗口。







15. 引导客户端将启动配置屏。这个向导将提示您为客户端输入一些基本配置选择。



16. 阅读并同意许可证协议。



点击许可证协议窗口中的 前进。

17. 配置防火墙。

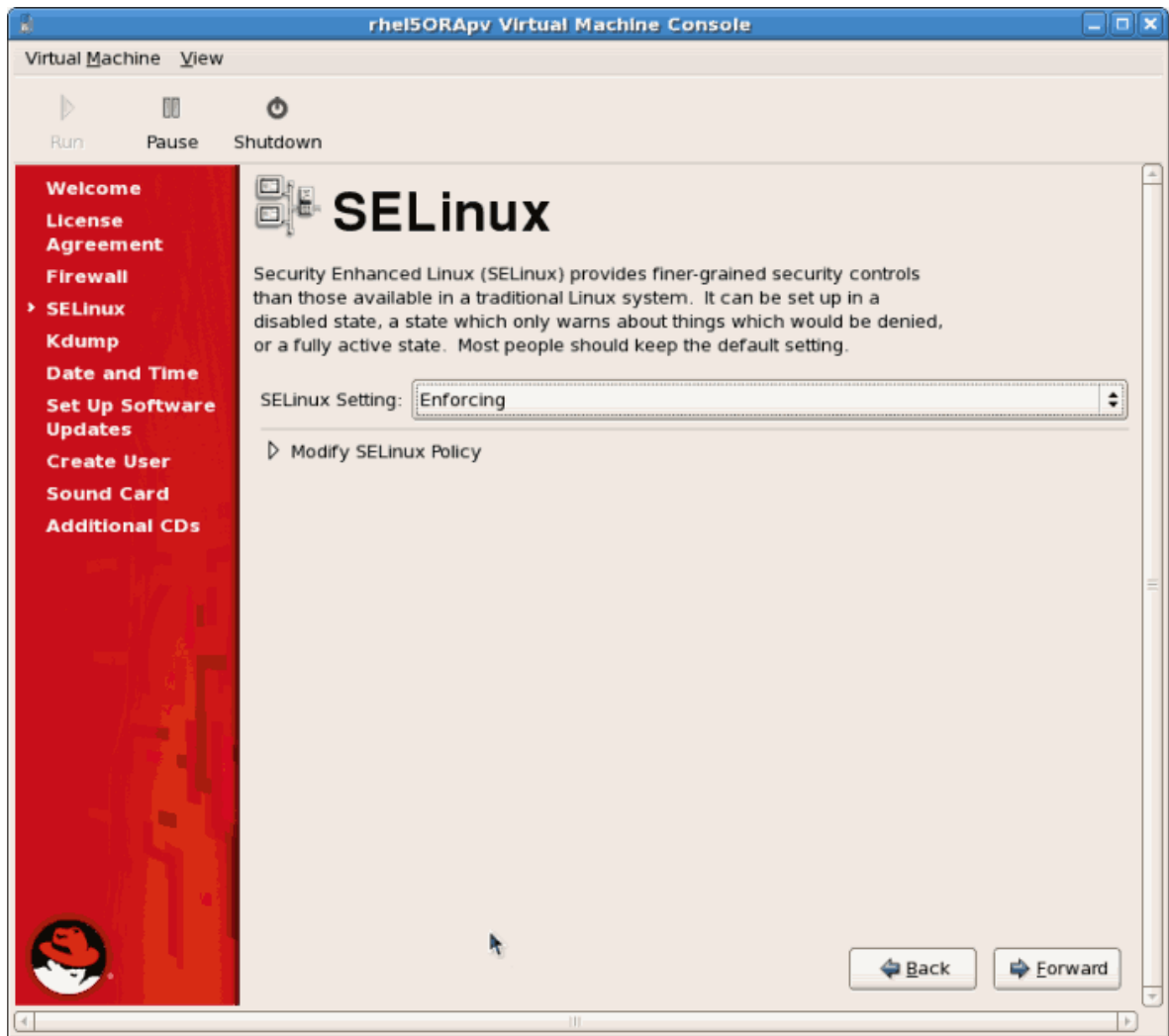


Click Forward to continue.

- 如果您选择禁用防火墙，则会提示您确认您的选择。点 确定 确认并继续。

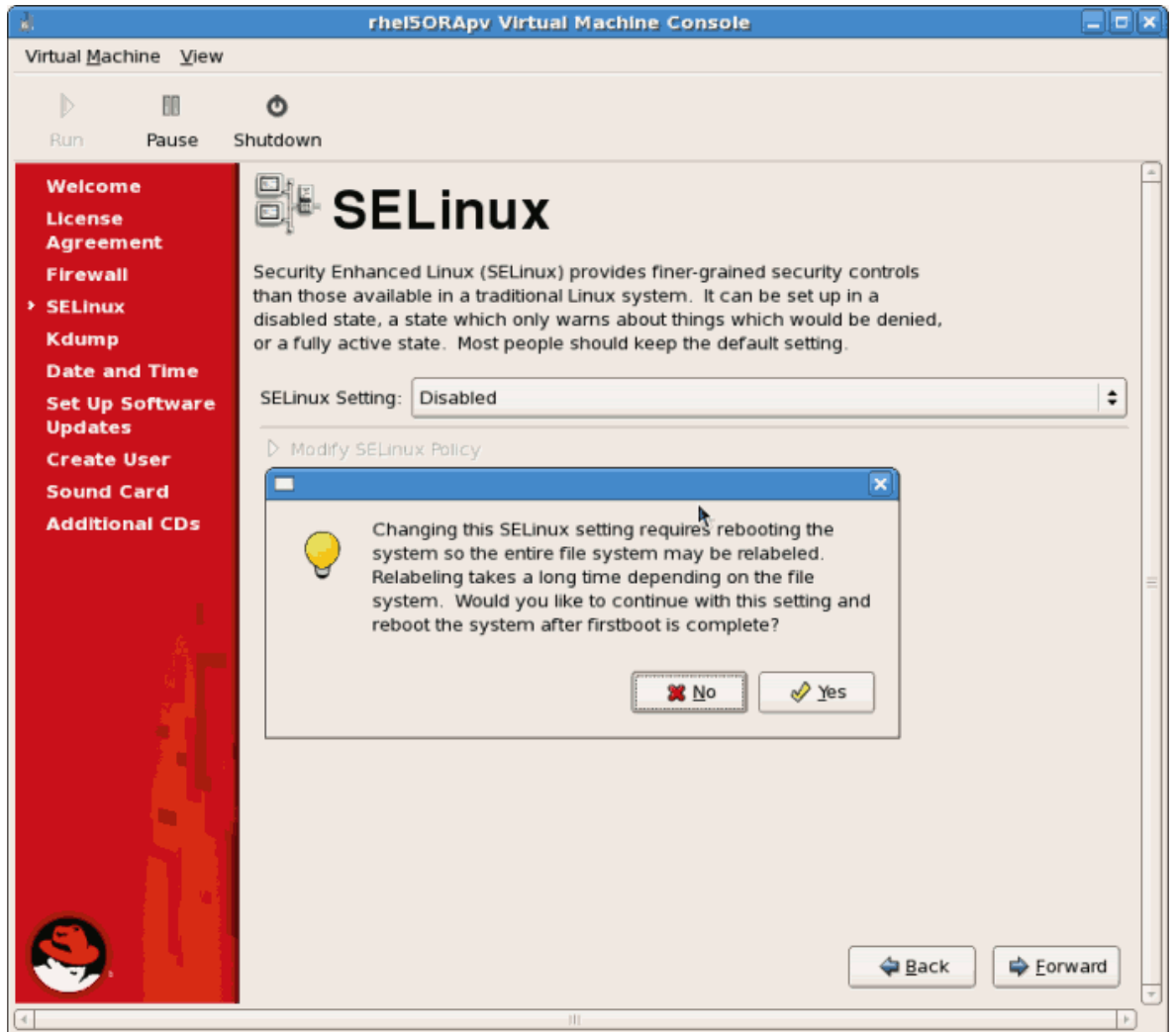


18. 配置 SELinux。我们强烈建议您在「enforcing」模式下运行 SELinux。您也可以选择在 permissive 模式下运行 SELinux 或者完全禁用 SELinux：



Click Forward to continue.

- 如果您选择禁用 SELinux，则会出现这个警告。点是 禁用 SELinux。



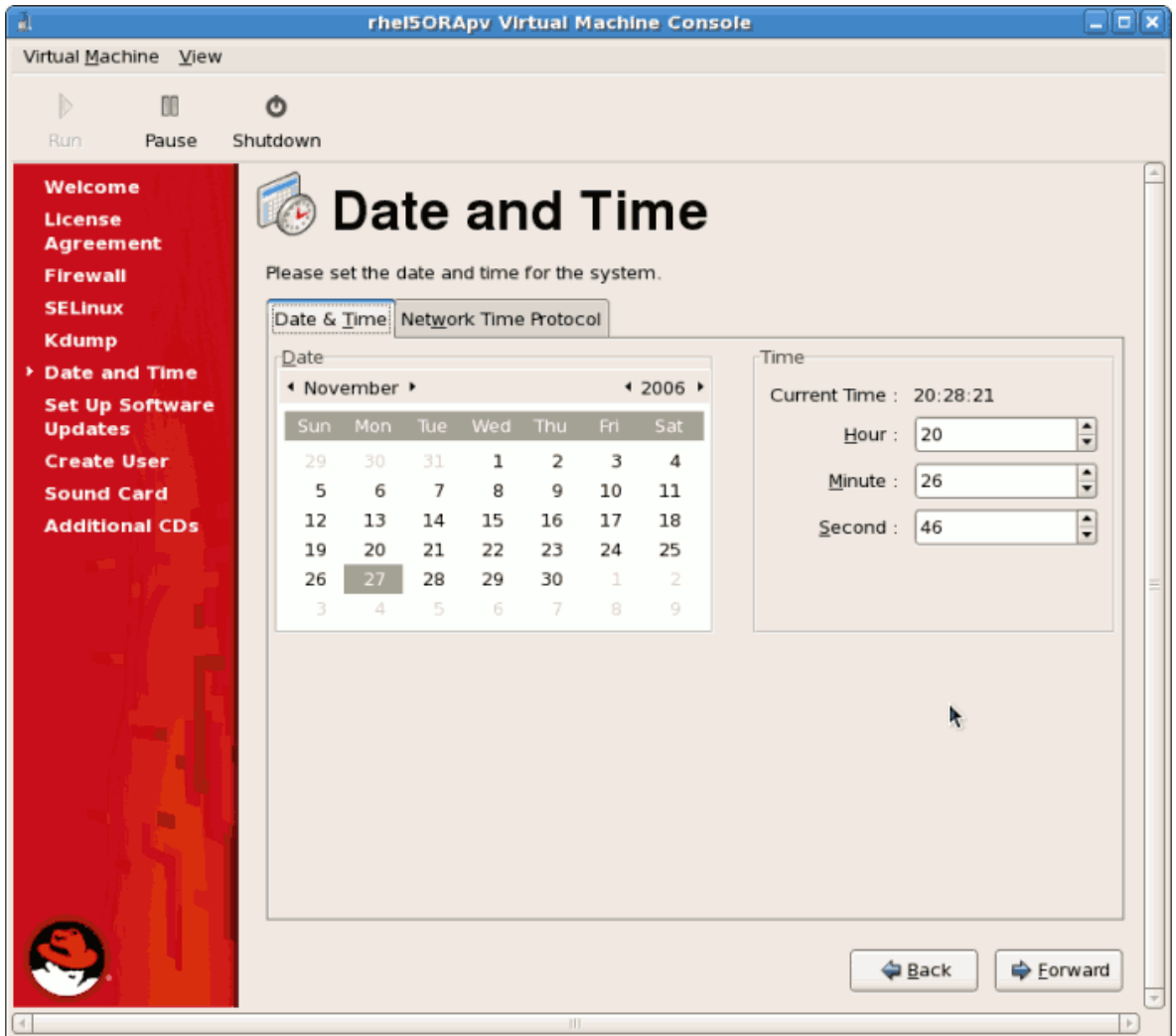
19. 需要时启用 kdump。



Click Forward to continue.

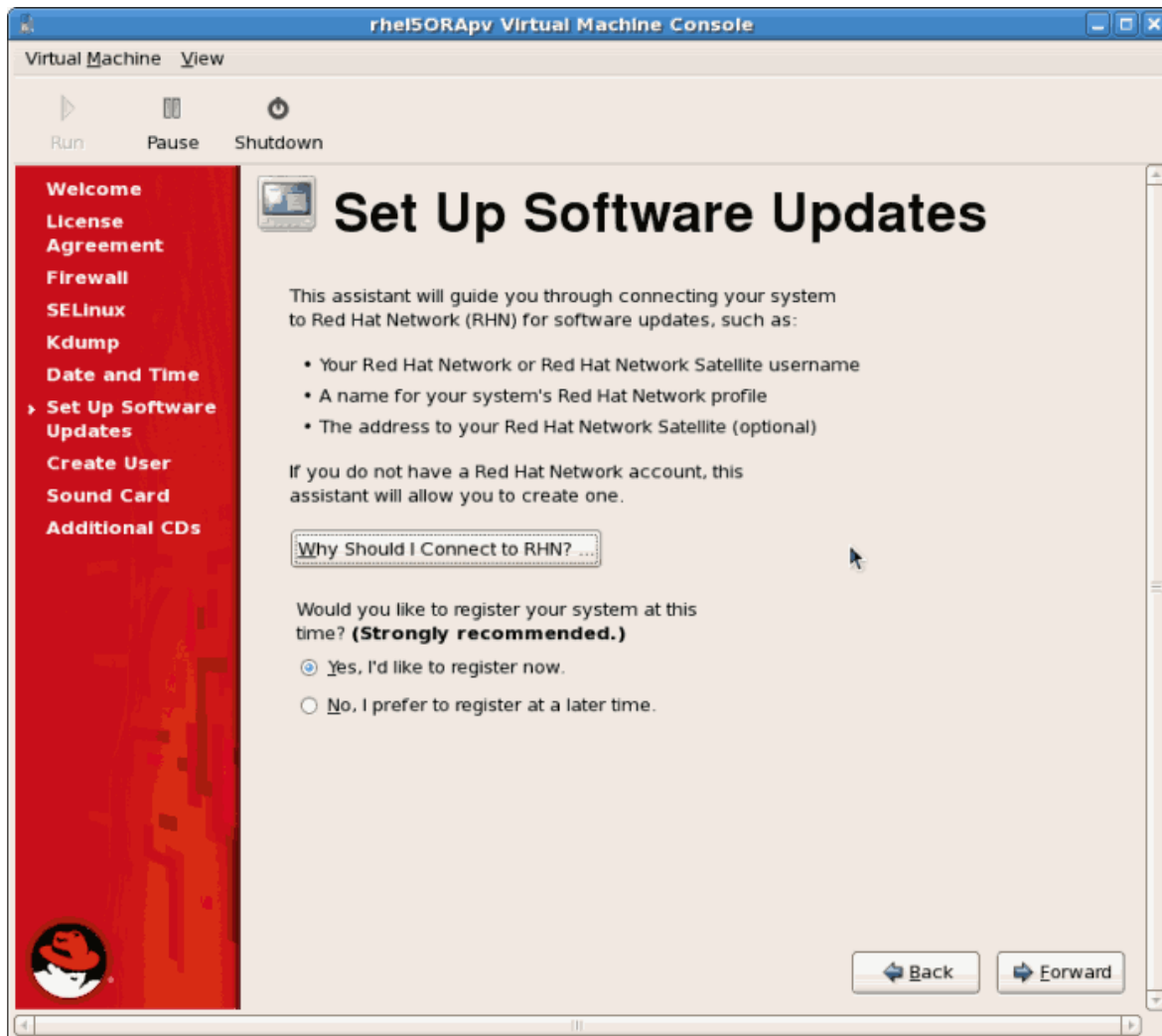
20. 确定为您的客户端正确设定了时间和日期。如果您安装了半虚拟客户端，则时间和日期应该与管理程序同步。





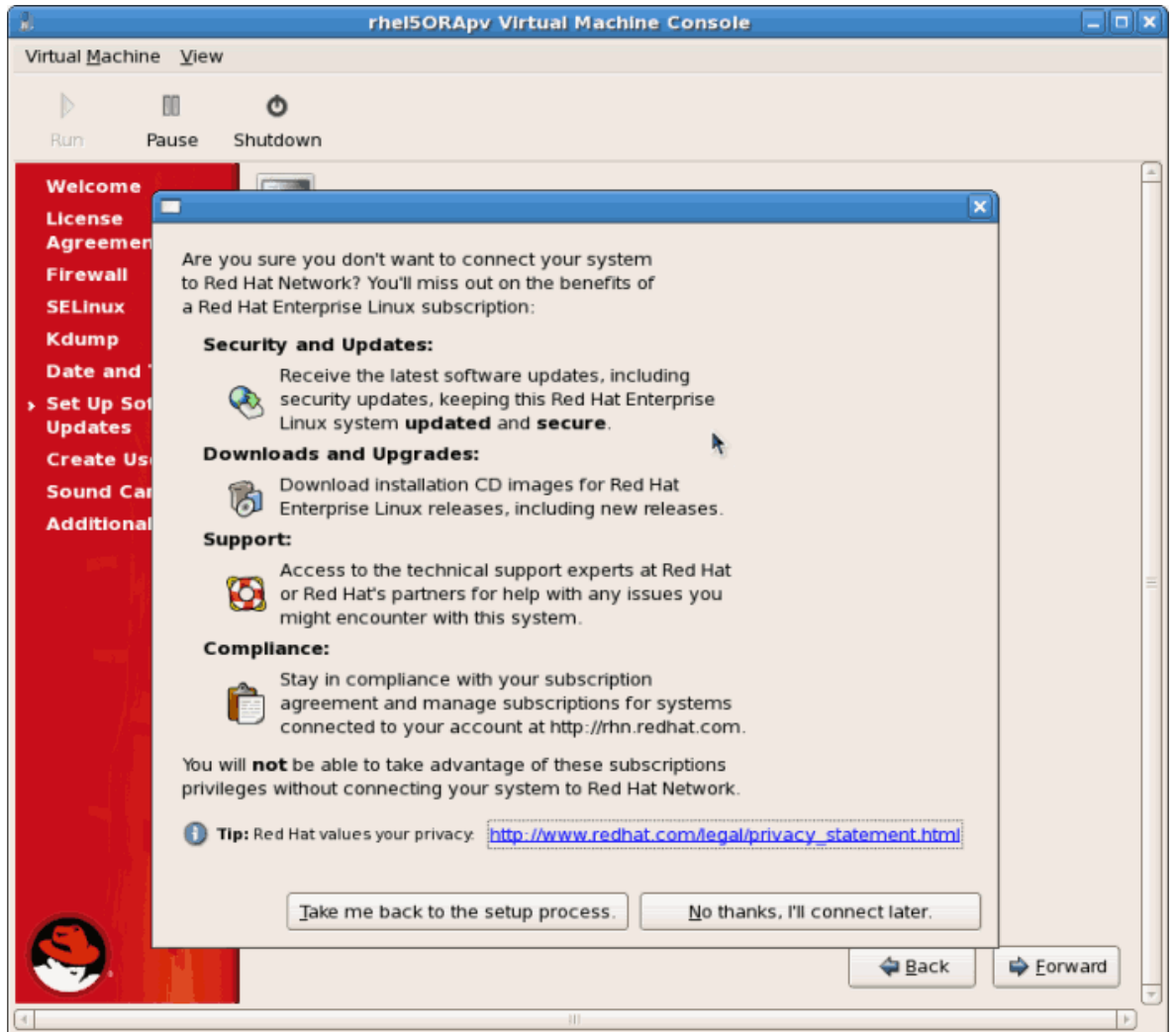
Click Forward to continue.

21. 设置软件更新。如果您有 fedora 网络订阅或者要尝试，请使用下面的页面在 RHN 中注册您新安装的客户端。

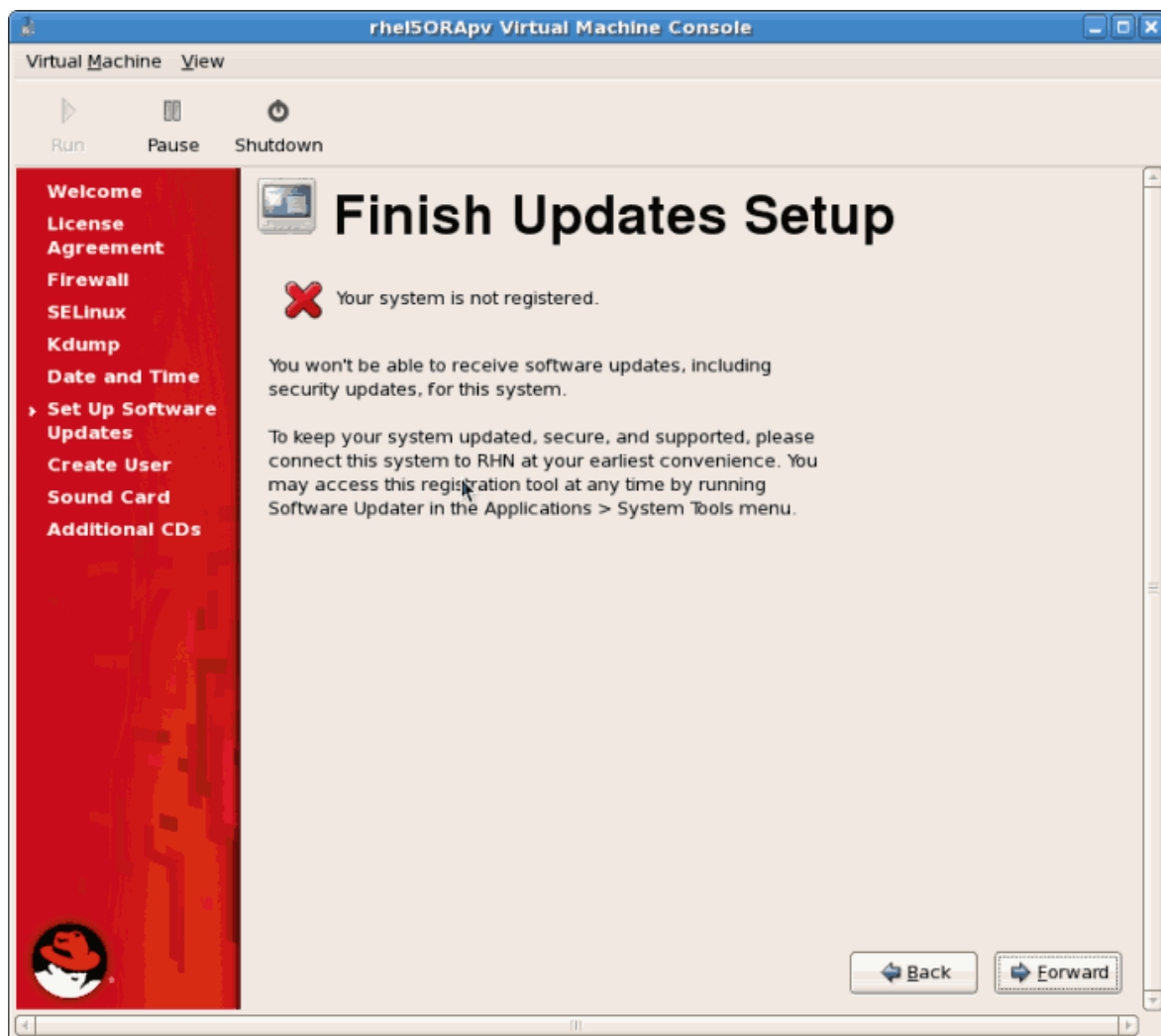


Click Forward to continue.

- a. 在 RHN 中确认您的选择。

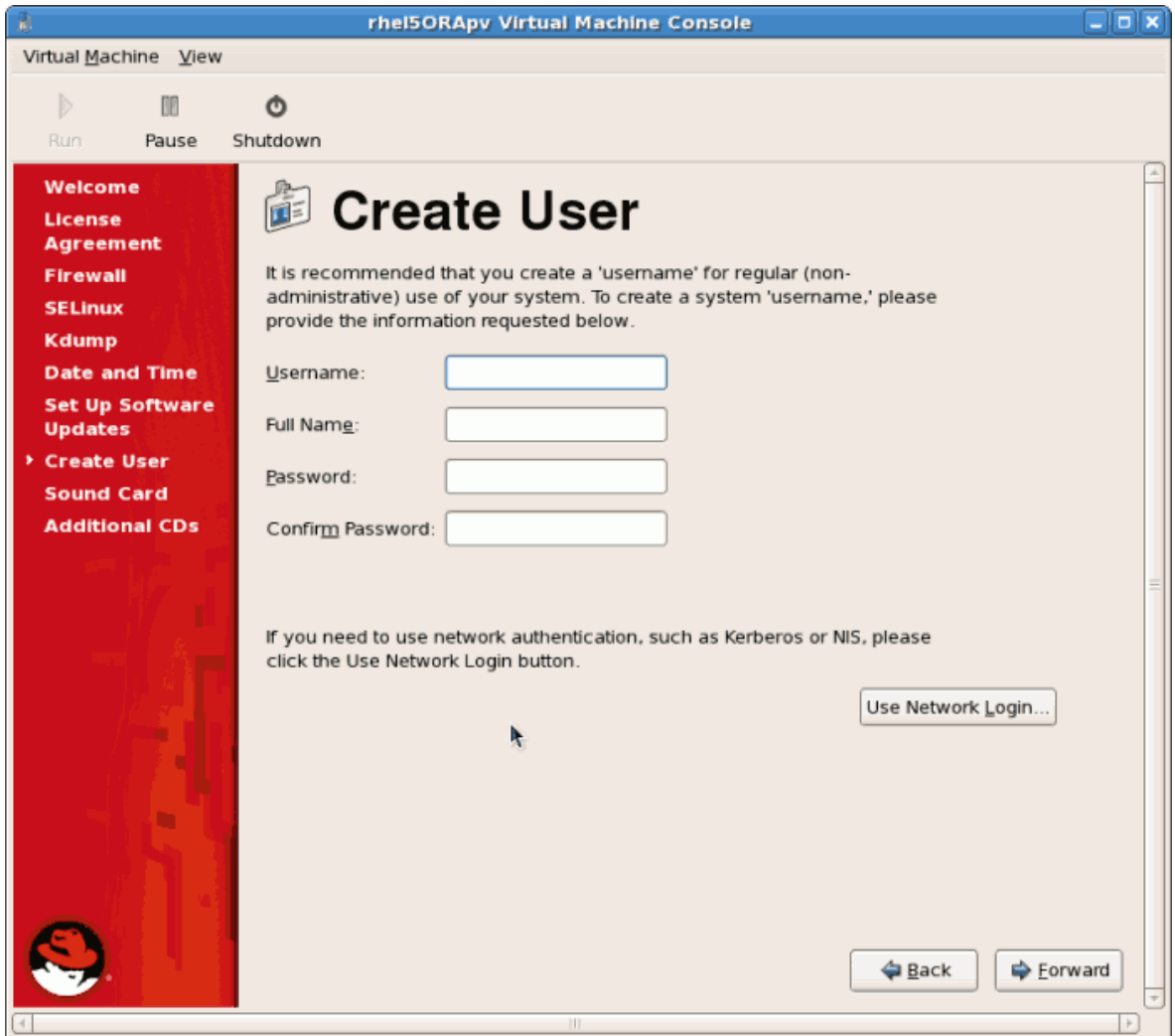


- b. 如果您这次没有选择 RHN，则在完成设置后，您可以看到一个或者多个页面。您将无法接受软件更新。



点击 前进 按钮。

22. 创建一个非根用户帐户。建议您为一般使用和加强的安全性创建一个非根用户。输入用户名、名称和密码。



点击 前进 按钮。

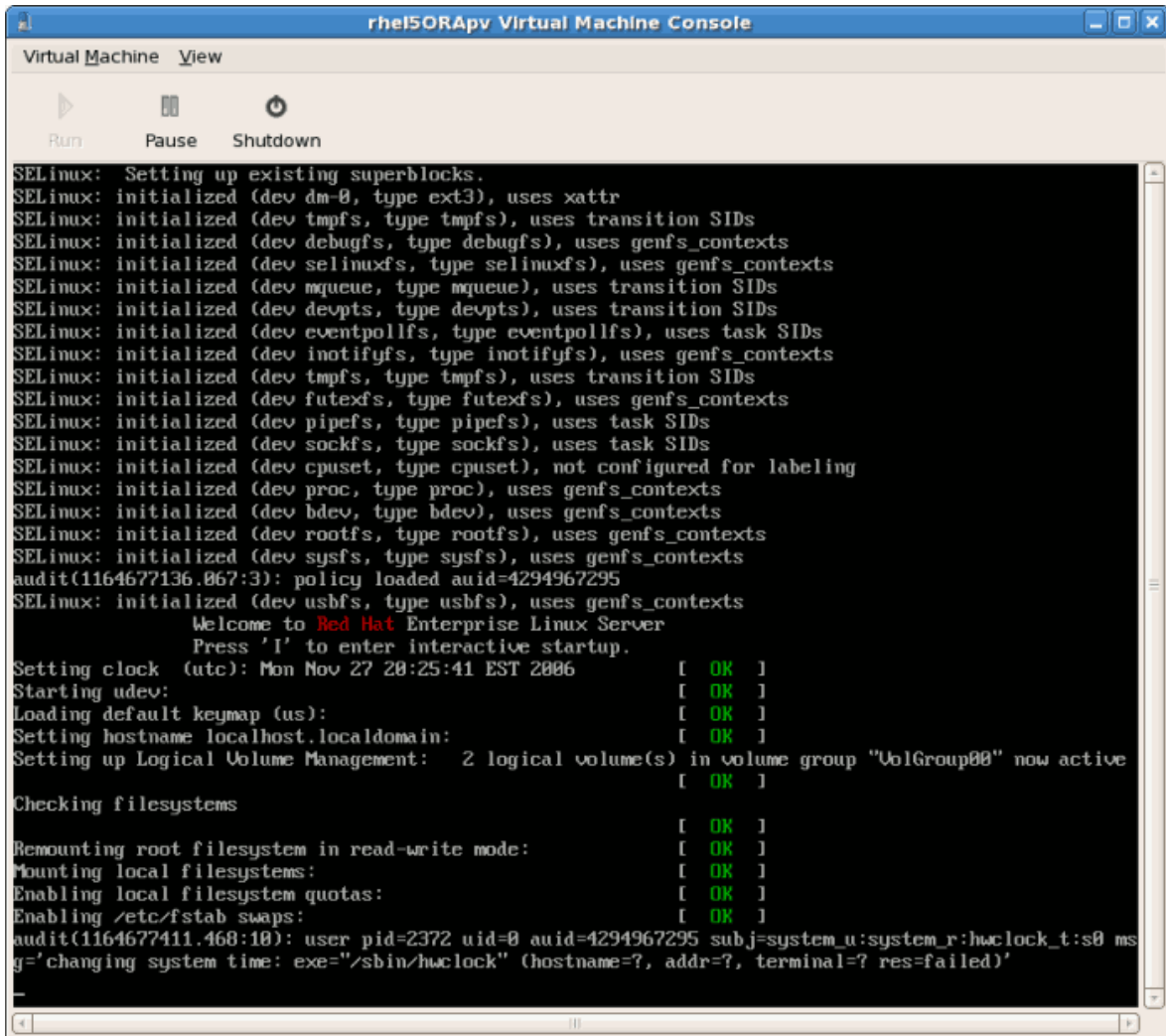
23. 如果侦测到一个声音设备且您需要音效，请校正它。完成该过程并点击 前进。



24. 在这个页面中您可以从光盘中安装任意附加软件软件包。通常不在此时安装任何额外软件，而在之后使用 yum 安装效率更高。点击 完成。

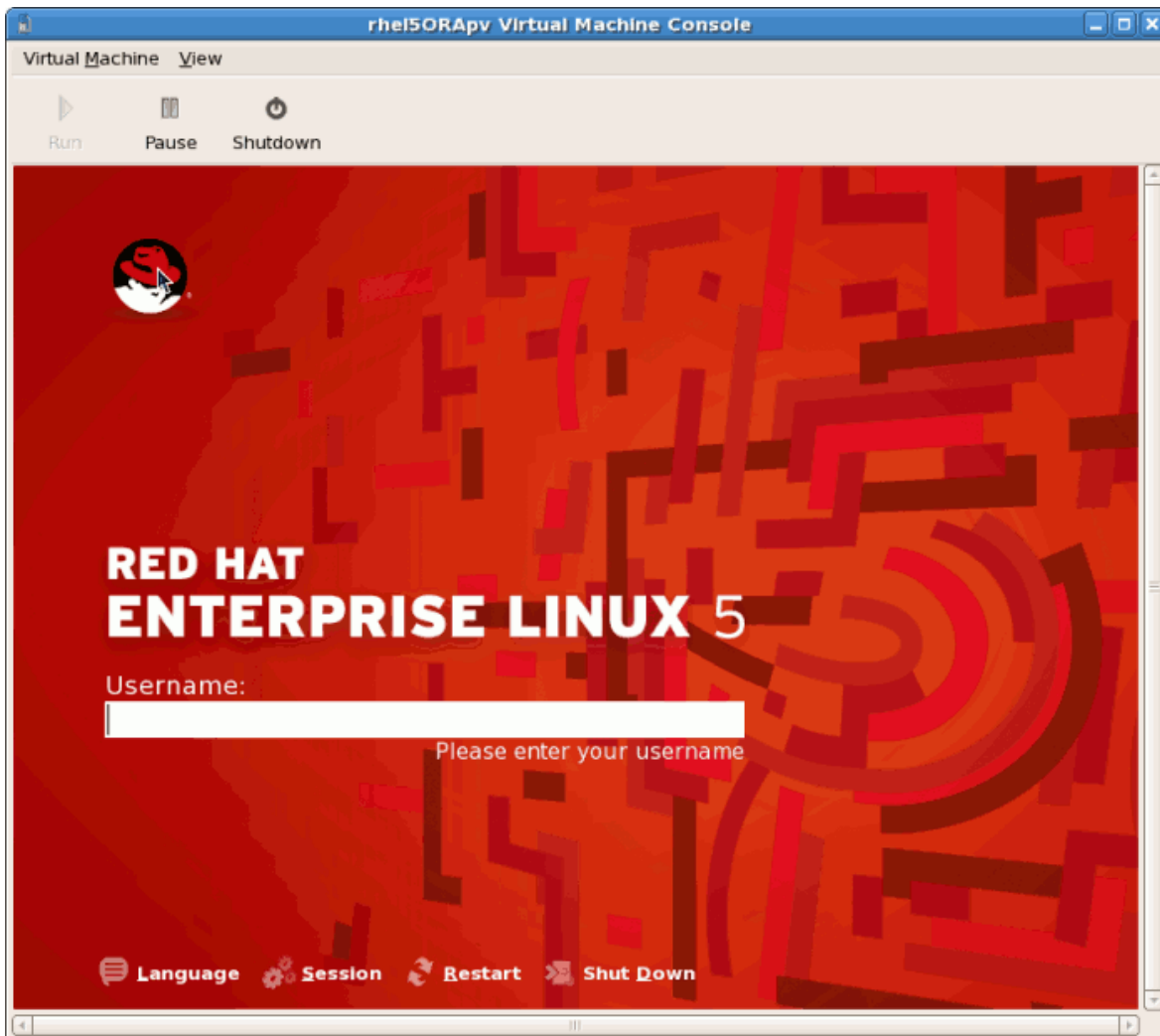


25. 客户端现在配置您更改的所有设定并继续引导过程。

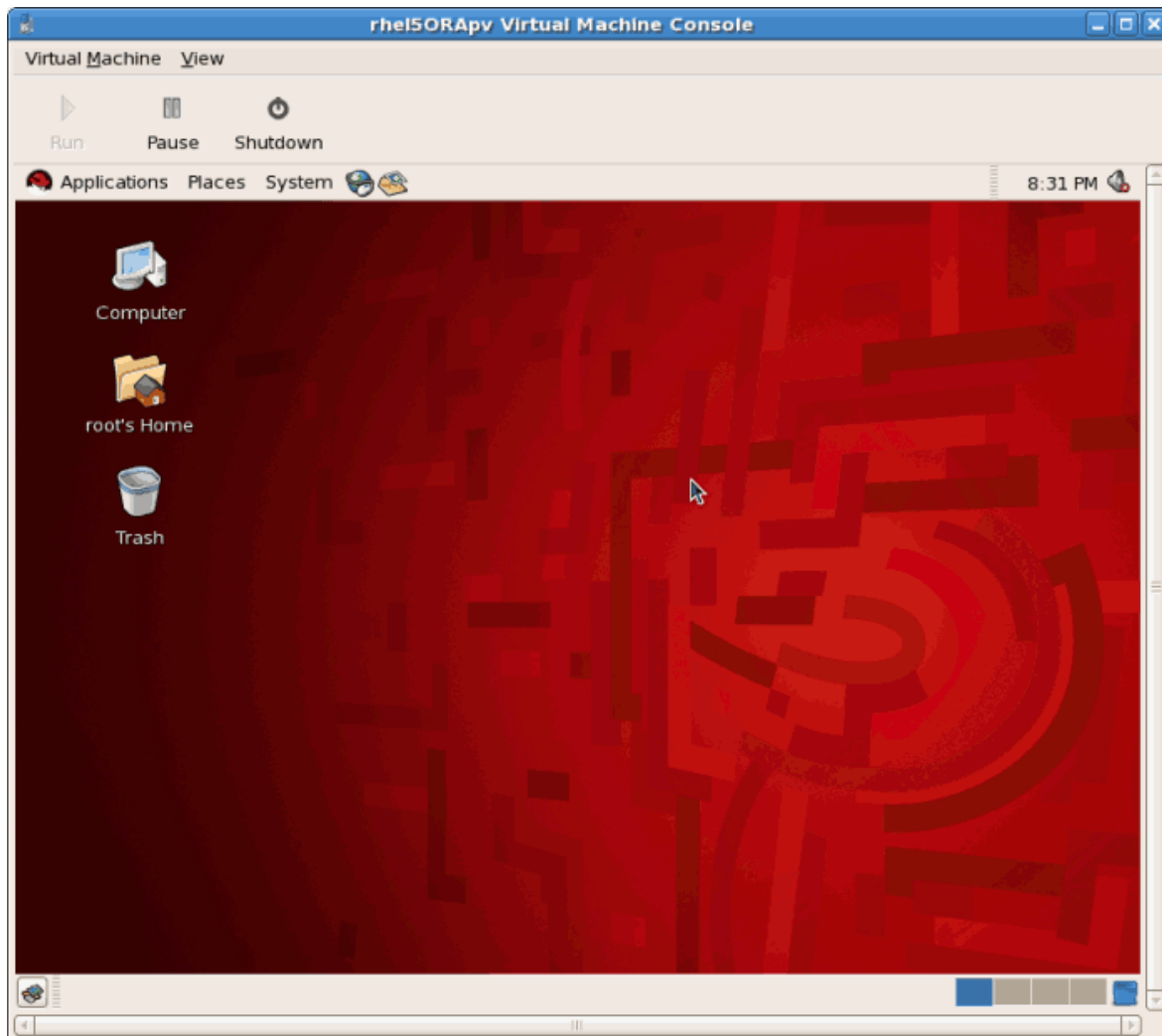


26. 显示运行红帽企业版 Linux 5 登录屏。使用前面步骤中创建的用户名登录。





27. 您已成功安装了半虚拟红帽企业版 Linux 5 客户端。



### 3.2. 将红帽企业版 Linux 作为全虚拟客户端安装

这部分论述如何安装全虚拟红帽企业版 Linux 5 客户端。

#### 过程 3.3. 使用 virt-manager 创建全虚拟红帽企业版 Linux 5 客户端

1. Open virt-manager  
Start virt-manager. Launch the Virtual Machine Manager application from the Applications menu and System Tools submenu. Alternatively, run the virt-manager command as root.
2. Select the hypervisor  
Select the hypervisor. If installed, select Xen or KVM. For this example, select KVM. Note that presently KVM is named qemu.

如果您还没有这样做，请连接到管理程序。打开「文件」菜单并选择「添加连接.....」选项，请参考 16.1 的“”。

选择管理程序连接后即可使用「新建」按钮。按「新建」按钮。

3. Start the new virtual machine wizard

Pressing the New button starts the virtual machine creation wizard.



Press Forward to continue.

4. Name the virtual machine

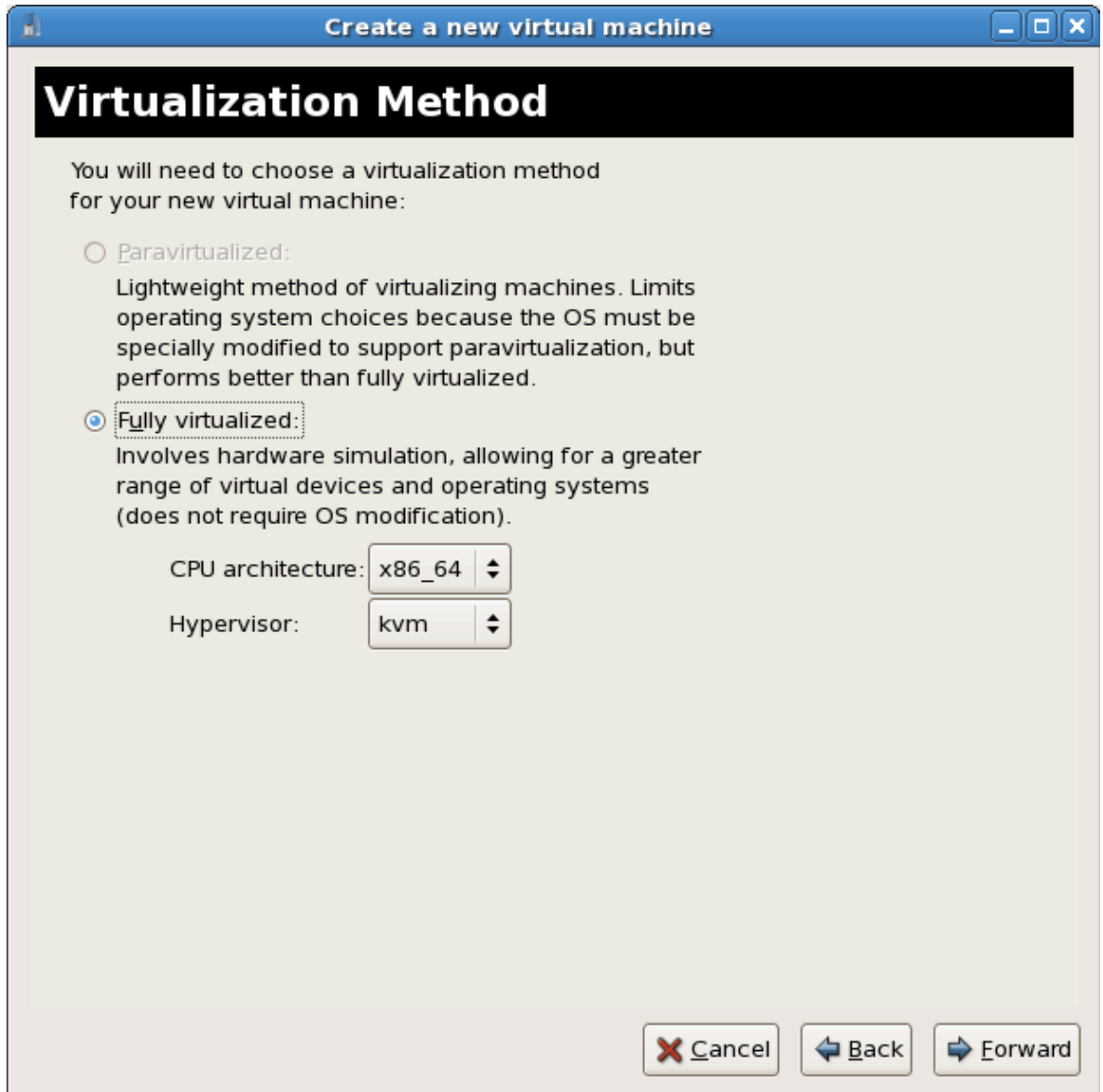
为您的虚拟客户端提供名称。不允许有标点和空白字符。



按 前进 继续。

5. Choose a virtualization method

为虚拟客户端选择虚拟化方法。请注意：您只能选择安装的虚拟化方法。如果您之前选择了 KVM 或者 Xen（004），您就必须使用已经选择的管理程序。这个示例使用 KVM 管理程序。

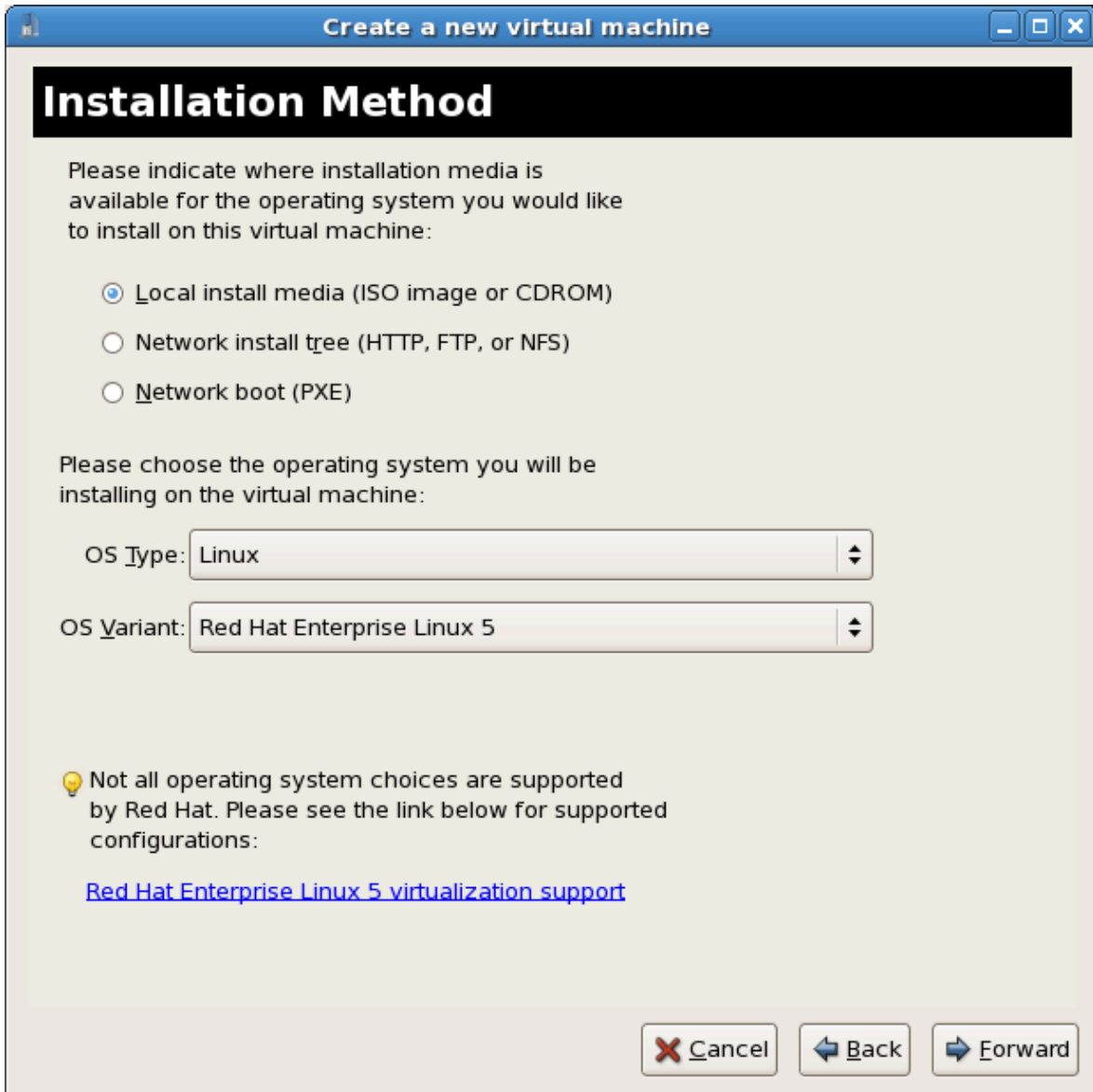


按 前进 继续。

6. Select the installation method

选择「本地安装介质」从光盘或者 ISO 映像安装；「网络安装树」从 HTTP、FTP 或者 NFS 服务器中安装；「网络引导」从 PXE 服务器中安装。

将「操作系统」类型设定为「Windows」；「操作系统变体」设定为「微软 Windows 2008」，如截屏中所示。



按 前进 继续。

7. Locate installation media

选择 ISO 映像位置或者 CD-ROM 或者 DVD 设备。这个示例使用红帽企业版 Linux 5 安装 DVD 的 ISO 文件映像。

- a. Press the Browse button.
- b. 搜索 ISO 文件位置并选择该 ISO 映像。按 打开 确认您的选择。
- c. 选择该文件并准备使用其进行安装。



按前进继续。

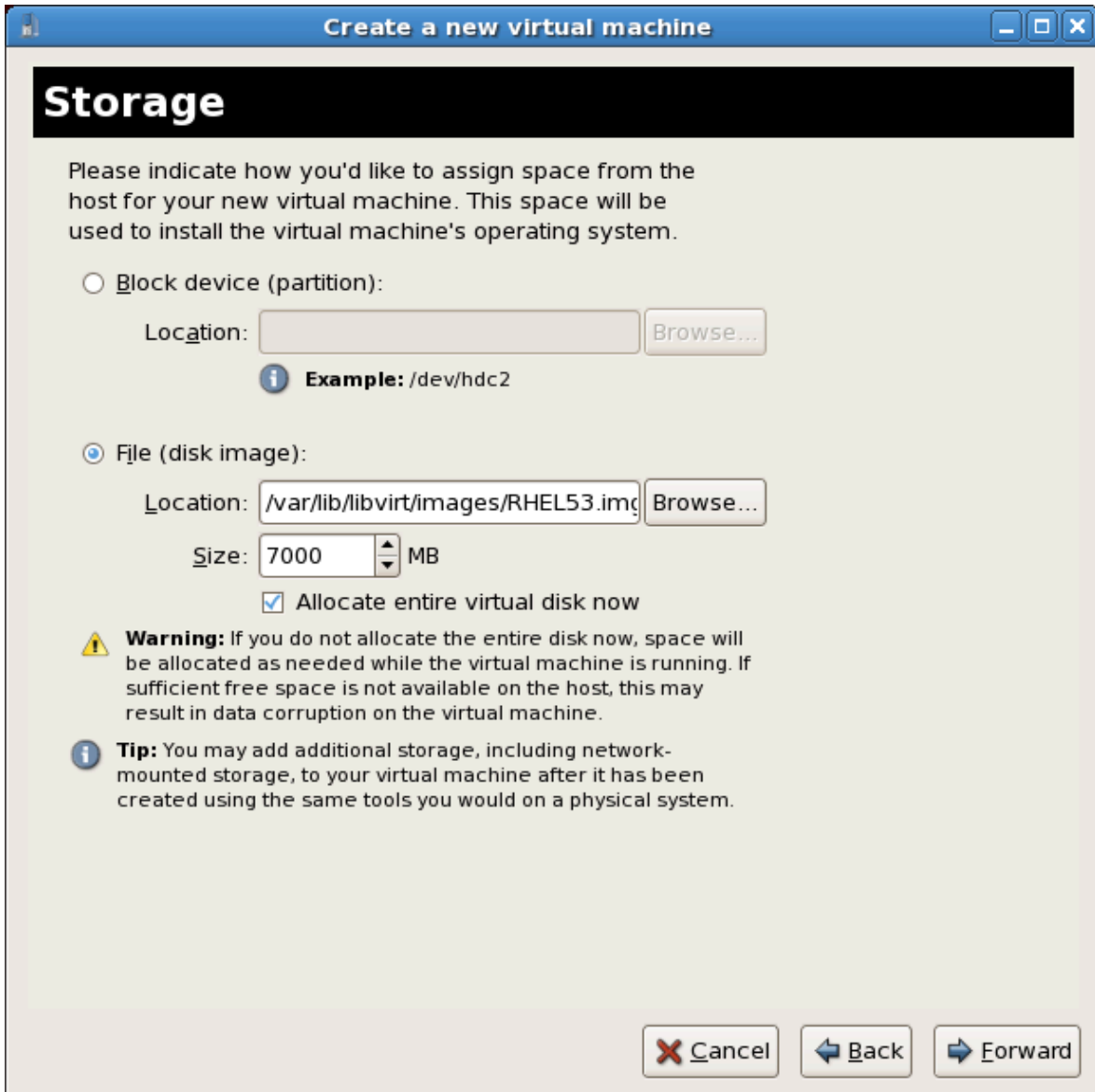


### Image files and SELinux


对于 ISO 映像文件和客户端存储映像，请使用 `/var/lib/libvirt/images/` 目录。其它位置可能需要额外配置 SELinux，详情请参考 [7.1 “SELinux 配置”](#)。

## 8. Storage setup

分配物理存储设备（「块设备」）或者基于文件的映像（「文件」）。基于文件的映像必须保存在 `/var/lib/libvirt/images/` 目录中。请为您的虚拟客户端分配足够的存储。为您的虚拟客户端及其所需应用程序分配足够的空间。



按 前进 继续。

 **迁移这个客户端**  
实时和离线迁移需要在共享的网络存储中安装该客户端。有关为客户端设置共享存储的信息请参考 5 。

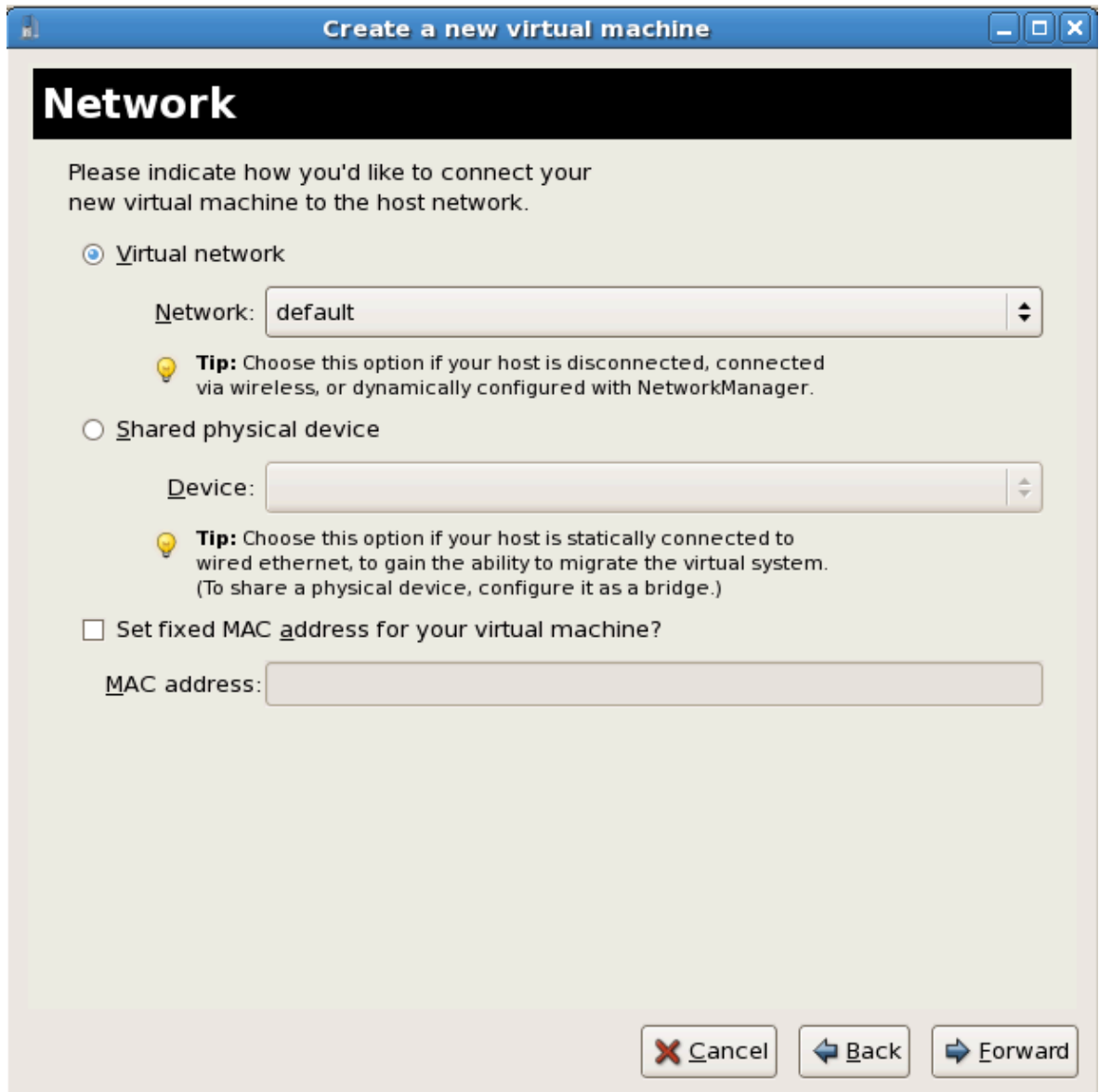
9. Network setup

Select either Virtual network or Shared physical device.

The virtual network option uses Network Address Translation (NAT) to share the default network device with the virtualized guest. Use the virtual network option for wireless networks.

The shared physical device option uses a network bond to give the virtualized guest full access to a network device.





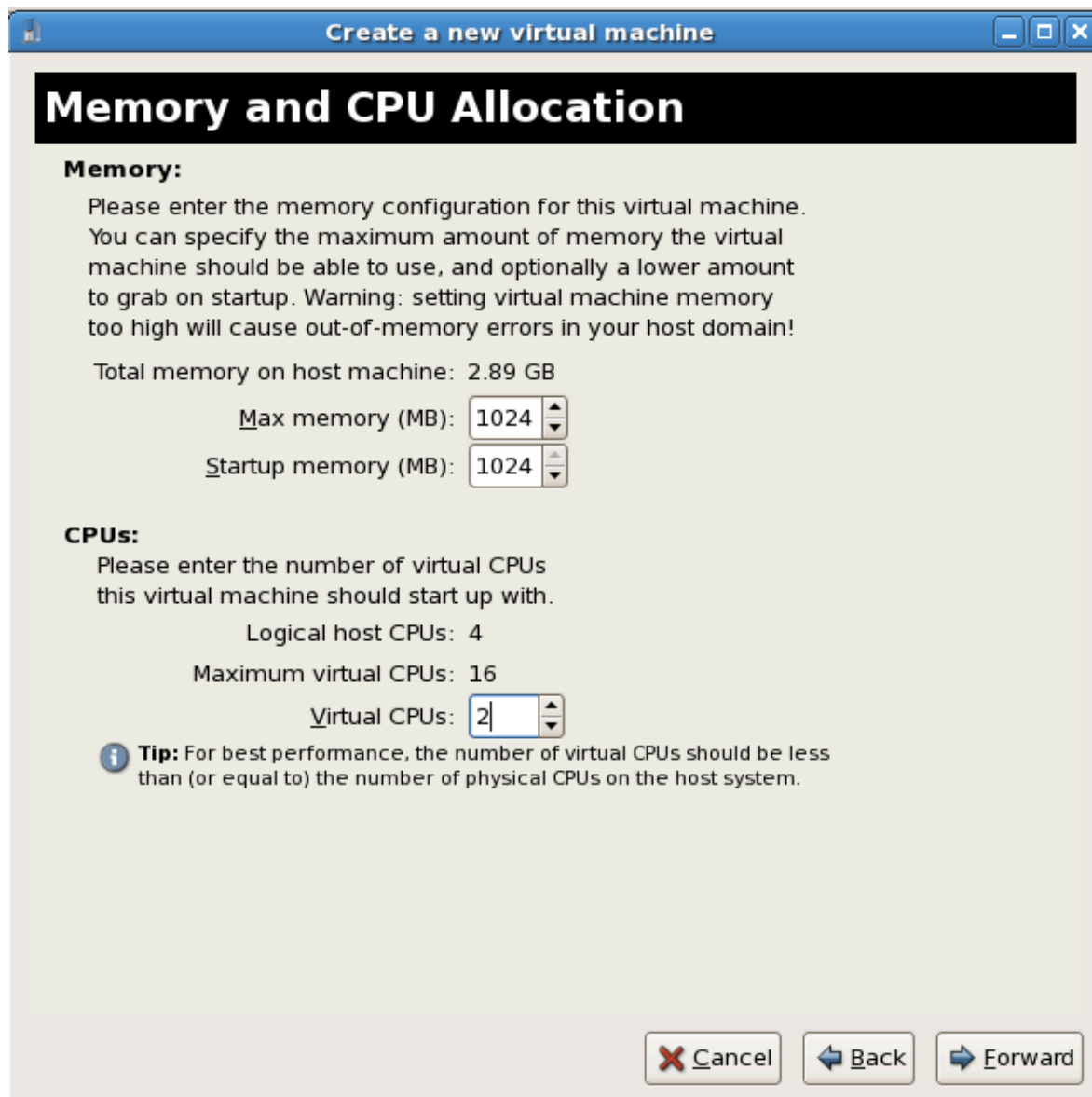
Press Forward to continue.

#### 10. Memory and CPU allocation

The Allocate memory and CPU window displays. Choose appropriate values for the virtualized CPUs and RAM allocation. These values affect the host's and guest's performance.

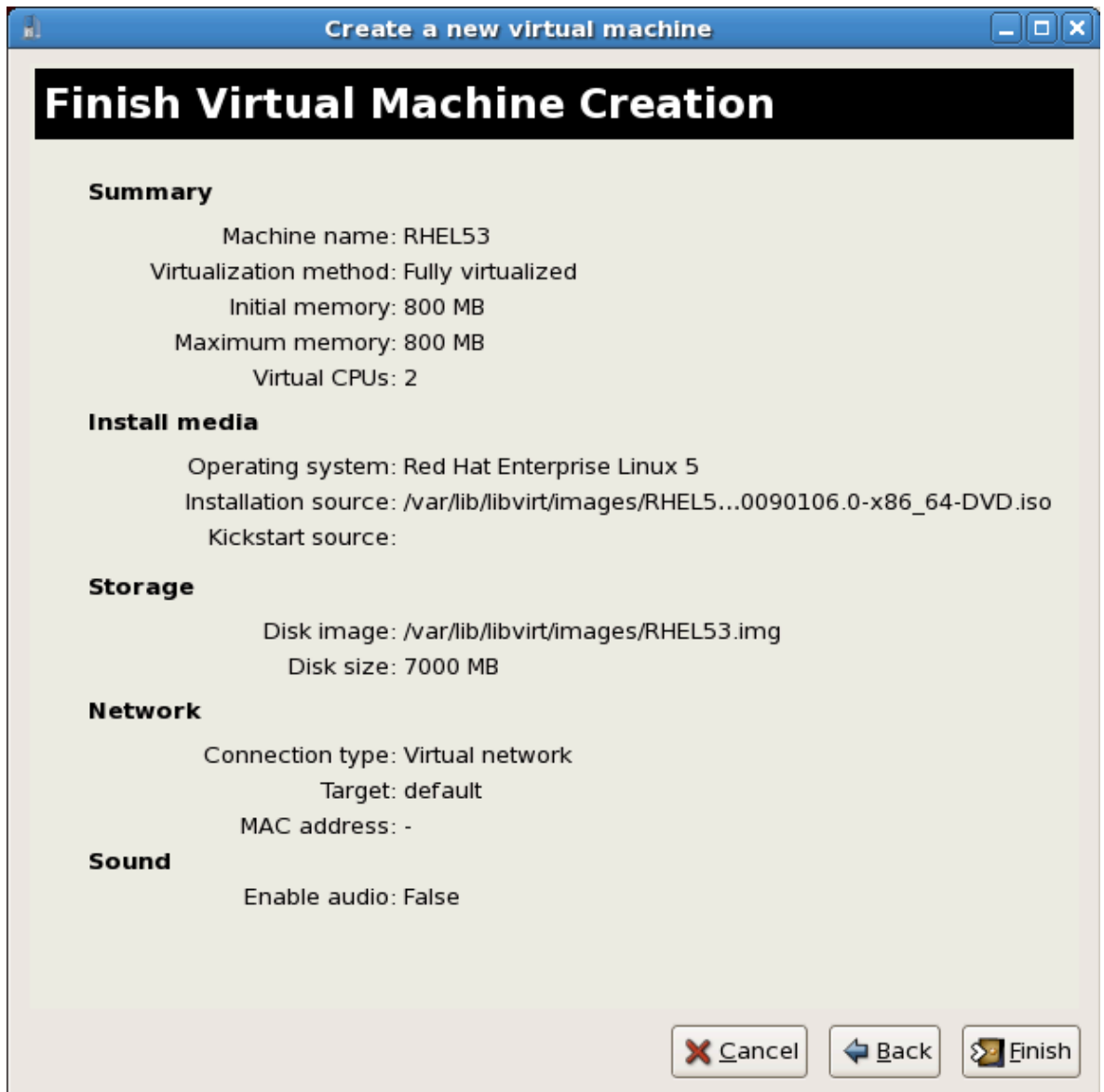
Virtualized guests require sufficient physical memory (RAM) to run efficiently and effectively. Choose a memory value which suits your guest operating system and application requirements, Windows Server 2008. Remember, guests use physical RAM. Running too many guests or leaving insufficient memory for the host system results in significant usage of virtual memory and swapping. Virtual memory is significantly slower causing degraded system performance and responsiveness. Ensure to allocate sufficient memory for all guests and the host to operate effectively.

Assign sufficient virtual CPUs for the virtualized guest. If the guest runs a multithreaded application assign the number of virtualized CPUs it requires to run most efficiently. Do not assign more virtual CPUs than there are physical processors (or hyper-threads) available on the host system. It is possible to over allocate virtual processors, however, over allocating has a significant, negative affect on guest and host performance due to processor context switching overheads.



Press Forward to continue.

- 11. Verify and start guest installation  
验证配置。



按 完成 启动客户端安装过程。

## 12. 安装 Linux

完成红帽企业版 Linux 5 安装步骤。安装顺序包含在 Linux 中，网址为 <http://redhat.com/docs>。

现在安装了全虚拟红帽企业版 Linux 5 客户端。

## 3.3. 将 Windows XP 作为全虚拟客户端安装

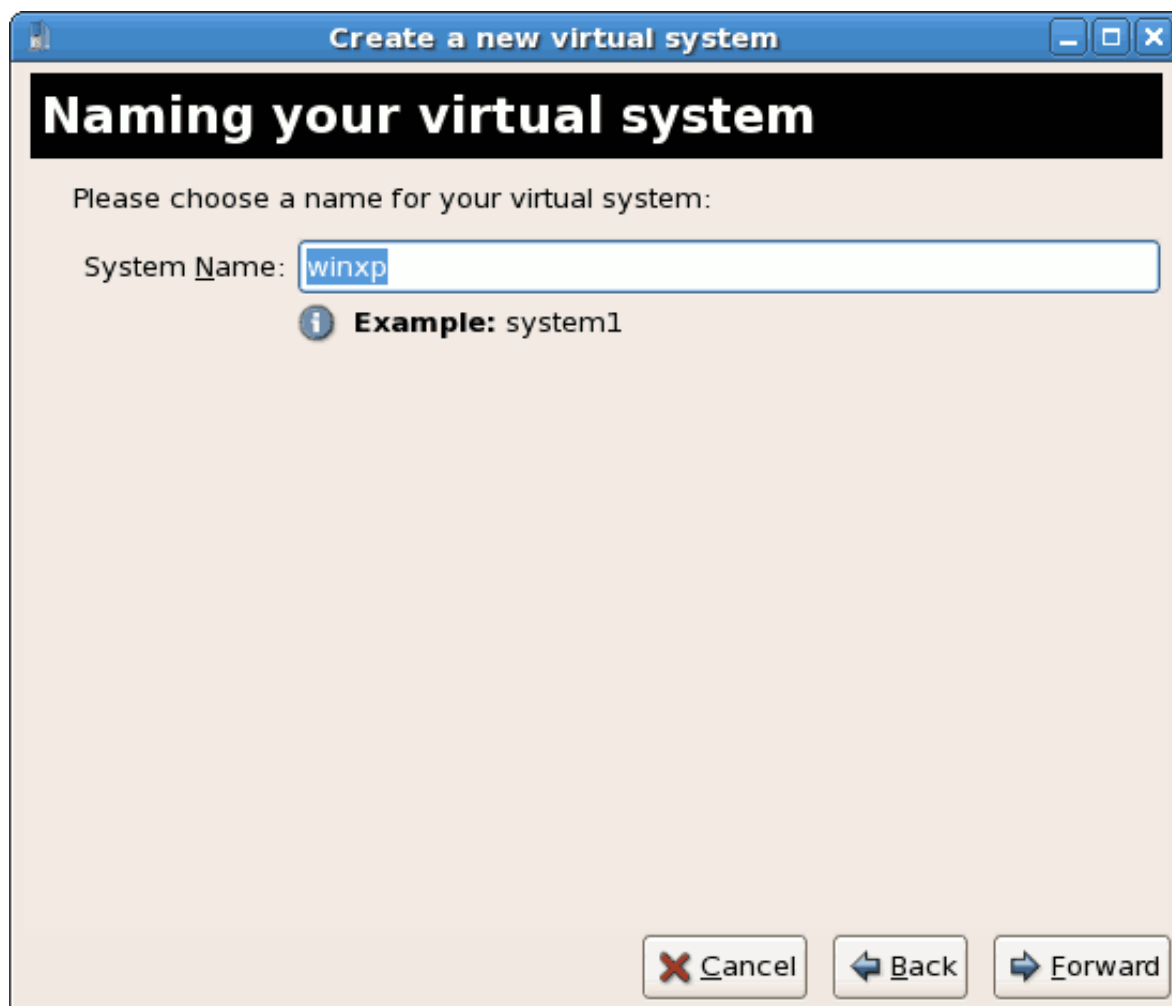
可将 Windows XP 作为全虚拟客户端安装。这部分论述了如何在 Linux 中将 Windows XP 作为全虚拟客户端安装。

进行此操作前请确定您可以作为根用户访问。

### 1. Starting virt-manager

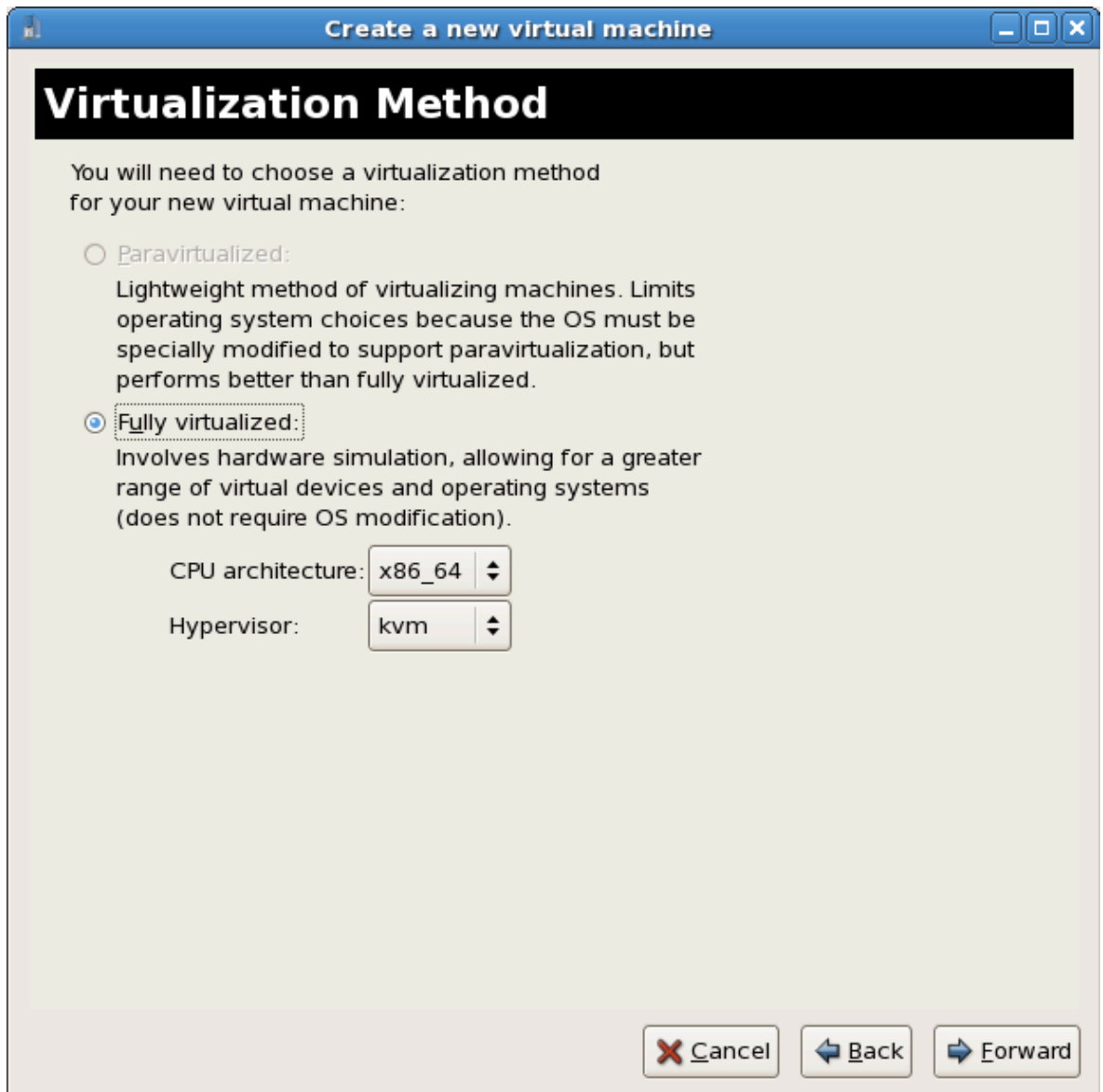
请打开「一样程序 > 系统工具 > 虚拟机管理器」。打开到该主机的连接（点击「文件 > 打开连接」）。点击 新建 按钮创建一个新虚拟机。

2. 为您的虚拟系统命名  
输入「系统名称」并点击 前进 按钮。



3. 选择虚拟化方法  
如果您之前选择 KVM 或者 Xen（步骤 00 1），您必须使用您选择的管理程序。这个示例使用 KVM 管理程序。

只能使用全虚拟化安装 Windows。



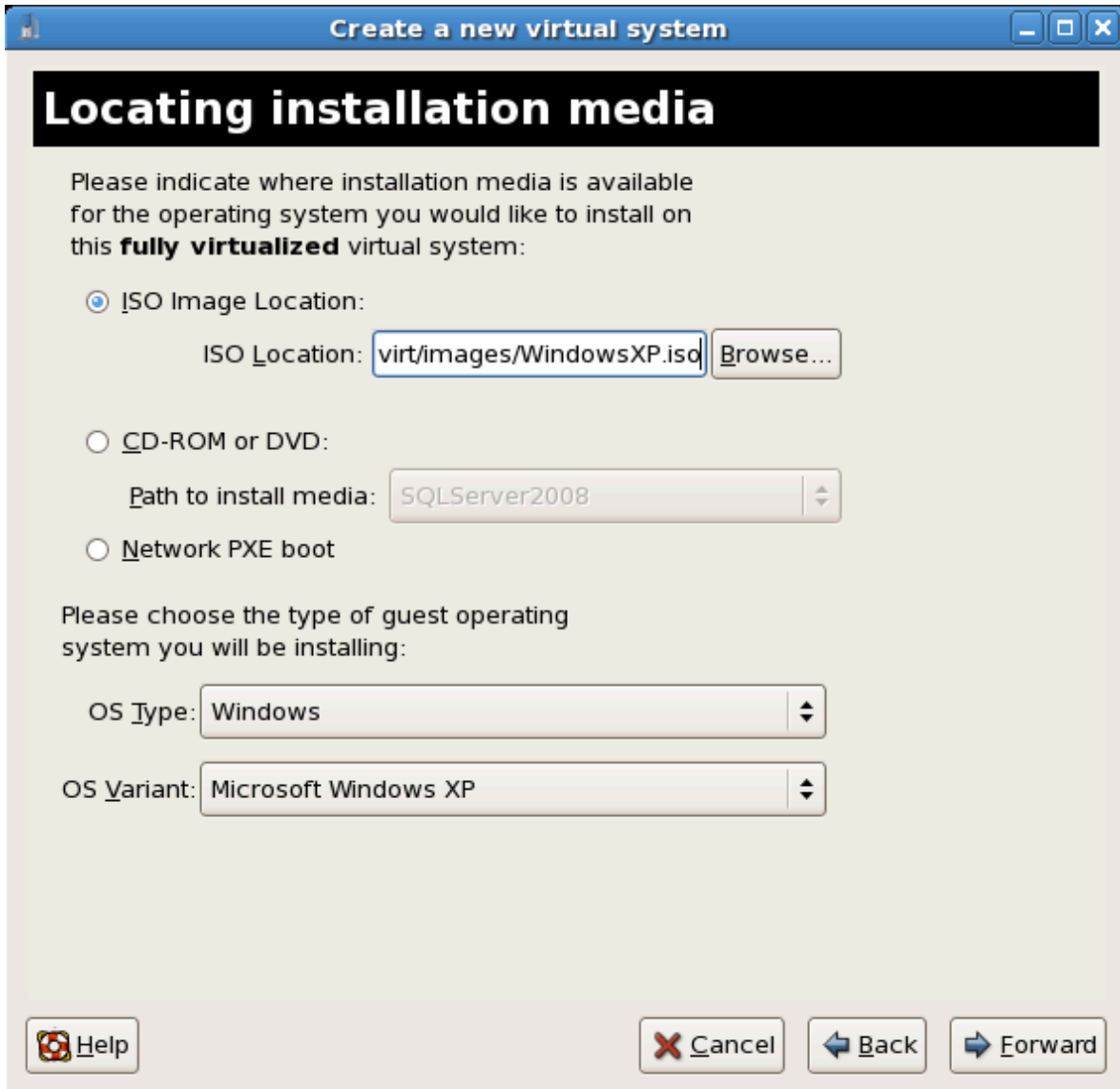
#### 4. 选择安装方法

这一步可让您指定操作系统的安装方法和类型。

如果使用 CD-ROM 或者 DVD 安装，请选择有 Windows 安装磁盘的设备。如果您选择「ISO 映像位置」，请输入 Windows 安装 .iso 映像的路径。

请在「操作系统类型」列表中选择「Windows」，在「操作系统变体」列表中选择「微软 Windows XP」。

本章不包括 PXE 安装。



Press Forward to continue.



### Image files and SELinux

对于 ISO 映像文件以及客户端存储映像，建议您使用 `/var/lib/libvirt/images/` 目录。其它任何位置可能需要对 SELinux 的额外配置，详情请参考 [7.1 “SELinux 配置”](#)。

5. The Assigning storage space window displays. Choose a disk partition, LUN or create a file based image for the guest storage.

在 Fedora 中对基于文件映像的规定是所有基于文件的客户端映像都应位于 `/var/lib/xen/images/` 目录中。SELinux 禁止在其它任何位置保存基于文件的映像。如果您以 enforcing 模式运行 SELinux，则请在 [7.1 “SELinux 配置”](#) 参考安装客户端的详情。

Your guest storage image should be larger than the size of the installation, any additional packages and applications, and the size of the guests swap file. The installation process will choose the size of the guest's swap file based on size of the RAM allocated to the guest.

Allocate extra space if the guest needs additional space for applications or other data. For example, web servers require additional space for log files.

**Create a new virtual system**

## Assigning storage space

Please indicate how you'd like to assign space on this physical host system for your new virtual system. This space will be used to install the virtual system's operating system.

Normal Disk Partition:

Partition:

**i** Example: /dev/hdc2

Simple File:

File Location:


File Size:

Allocate entire virtual disk now?

**Warning:** If you do not allocate the entire disk at VM creation, space will be allocated as needed while the guest is running. If sufficient free space is not available on the host, this may result in data corruption on the guest.

**i** **Tip:** You may add additional storage, including network-mounted storage, to your virtual system after it has been created using the same tools you would on a physical system.

Choose the appropriate size for the guest on your selected storage type and click the Forward button.

 **注意**

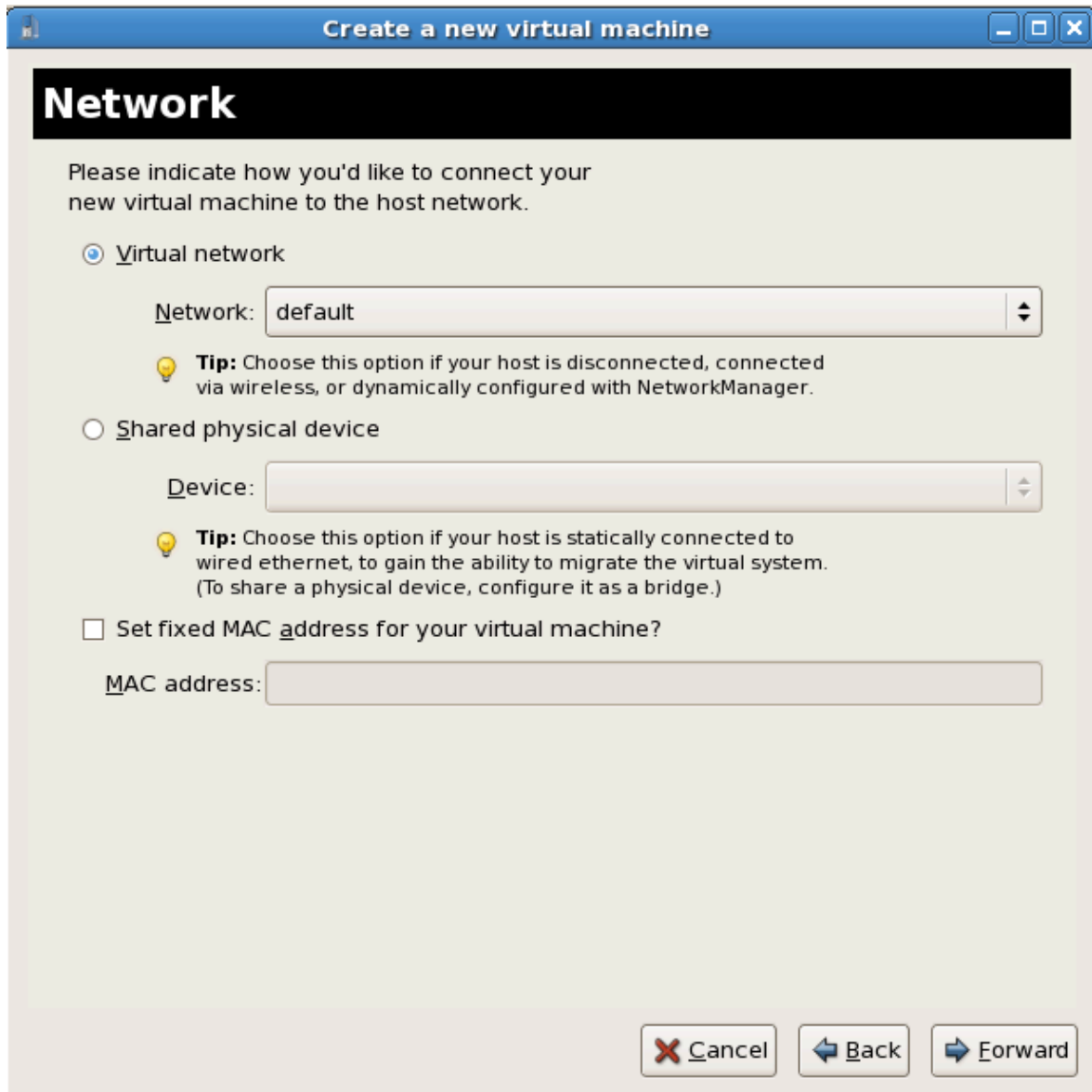
建议您使用默认目录保存虚拟机映像，默认目录为 `/var/lib/libvirt/images/`。如果您要使用不同的位置（比如在本示例中使用 `/images/`），请确定将其添加到 SELinux 策略中并在您继续安装前对其进行重新标记（后面会介绍如何修改 SELinux 策略）。

## 6. Network setup

Select either Virtual network or Shared physical device.

The virtual network option uses Network Address Translation (NAT) to share the default network device with the virtualized guest. Use the virtual network option for wireless networks.

The shared physical device option uses a network bond to give the virtualized guest full access to a network device.



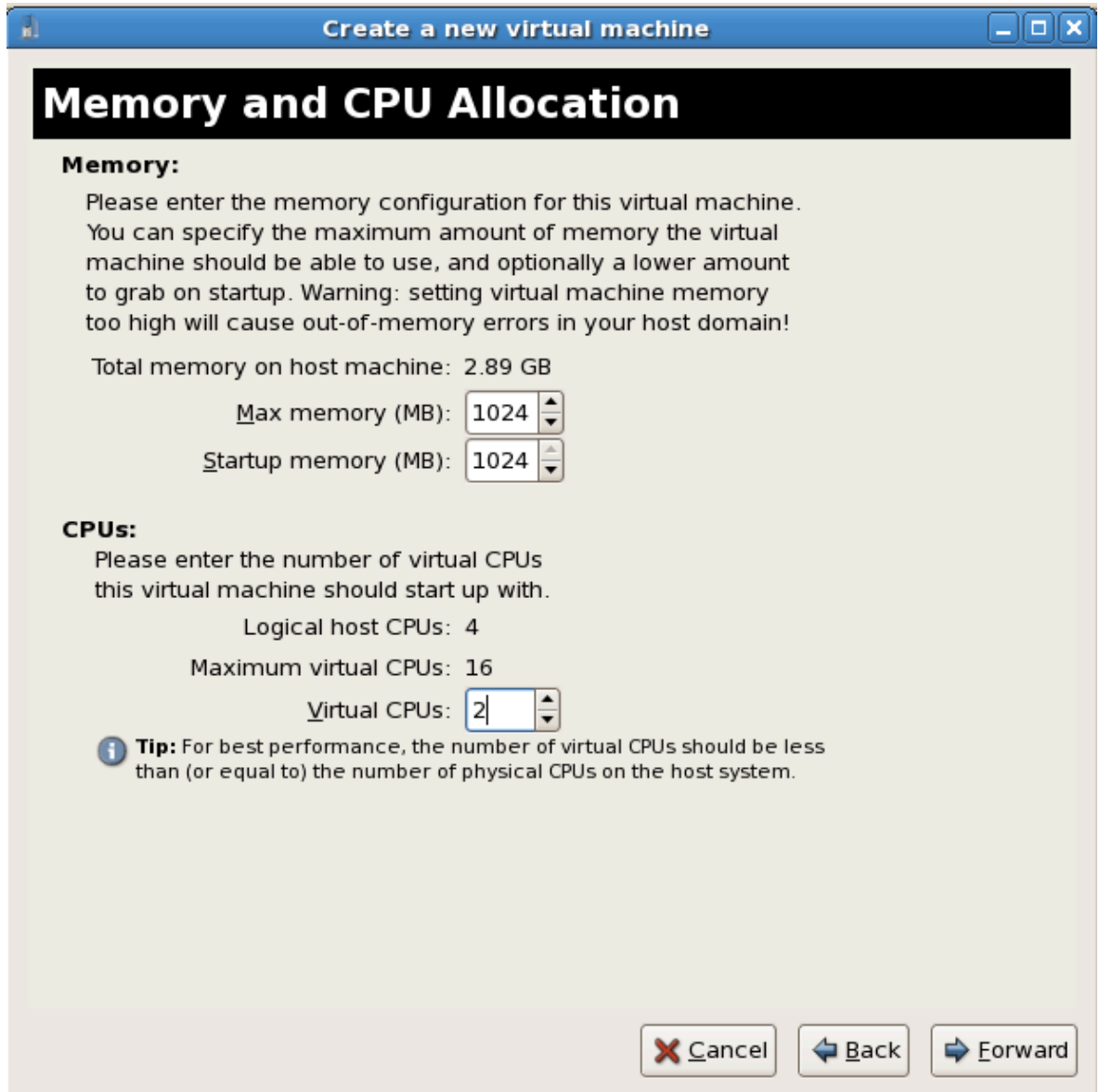
Press Forward to continue.

7. The Allocate memory and CPU window displays. Choose appropriate values for the virtualized CPUs and RAM allocation. These values affect the host's and guest's performance.

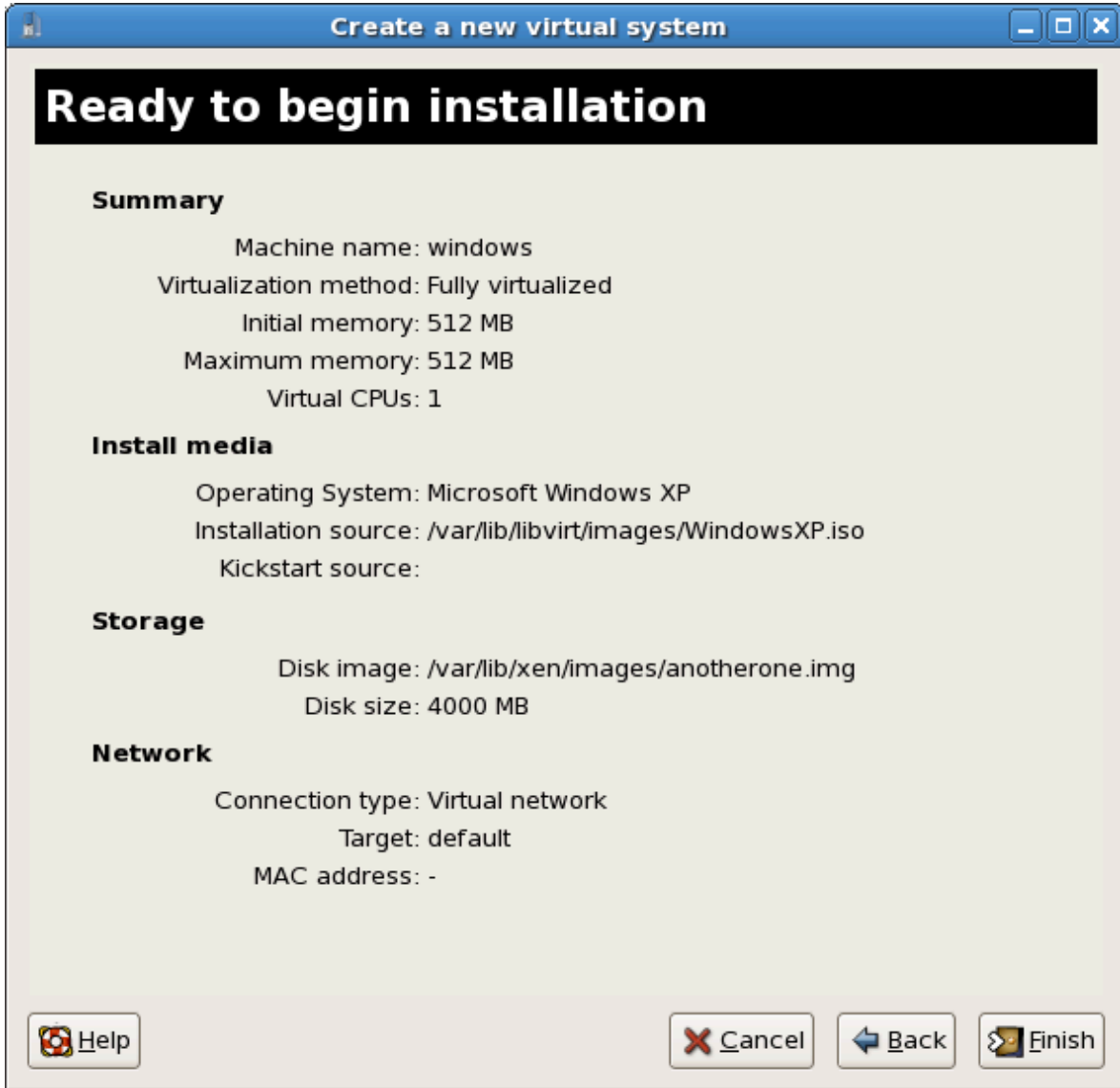
虚拟客户端需要足够的内存（RAM）才可有效运行。选择适合您客户端操作系统和应用程序需要的内存值。大多数操作系统需要至少 512MB 内存才可工作。请记住：客户端使用的是物理 RAM。运行太多客户端或者留给主机系统的内存不足将导致过量使用虚拟内存和交换。虚拟内存速度要慢很多从而降低系统性能和反应速度。确定您为所有客户端和主机分配了足够的内存以便其有效操作。

Assign sufficient virtual CPUs for the virtualized guest. If the guest runs a multithreaded application assign the number of virtualized CPUs it requires to run most efficiently. Do not assign more virtual CPUs than there are physical processors (or hyper-threads) available on the host system. It is possible to over allocate virtual processors, however, over allocating has a significant, negative affect on guest and host performance due to processor context switching overheads.

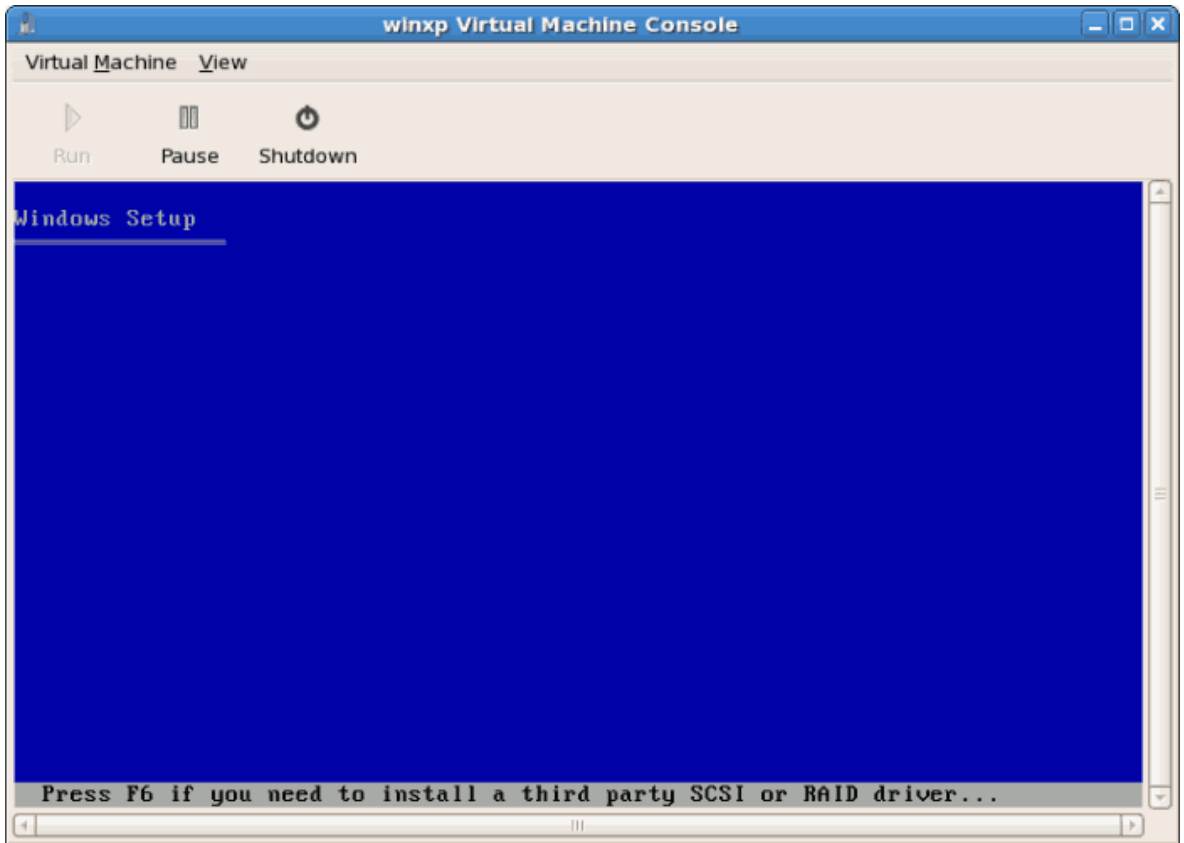




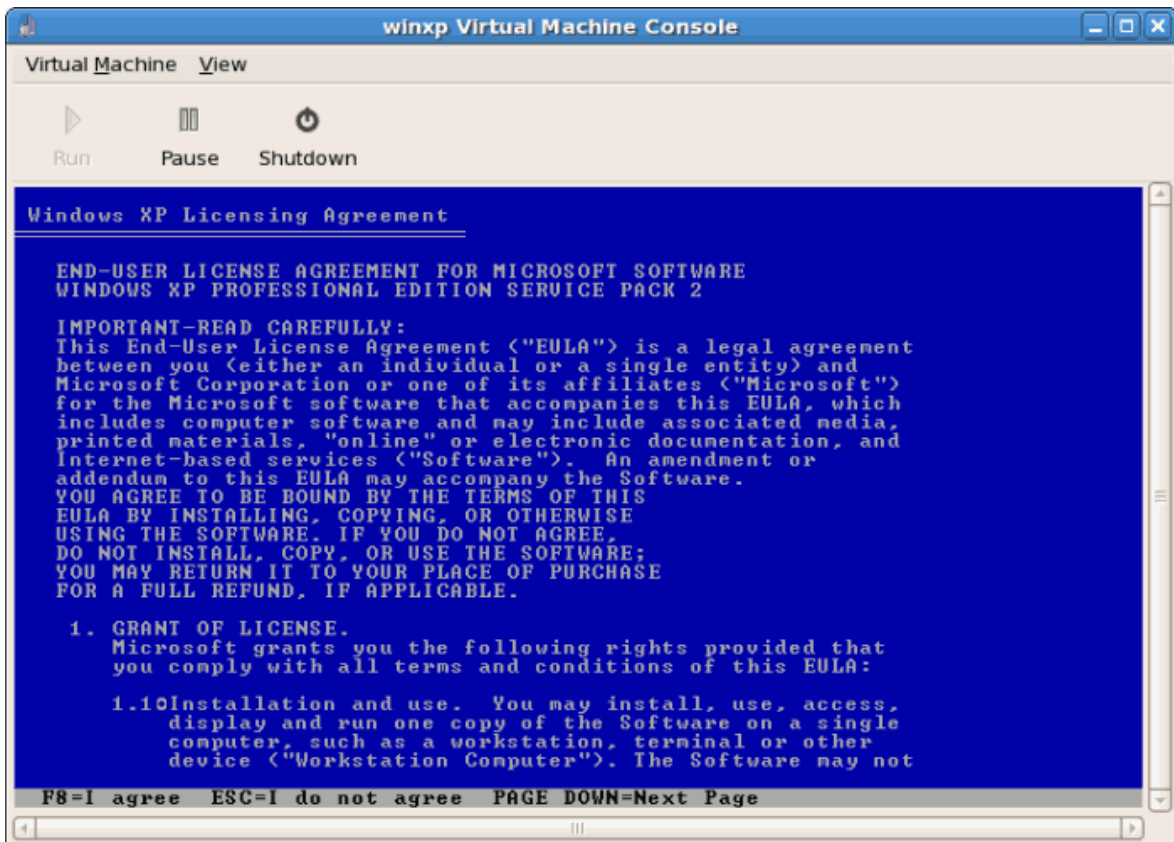
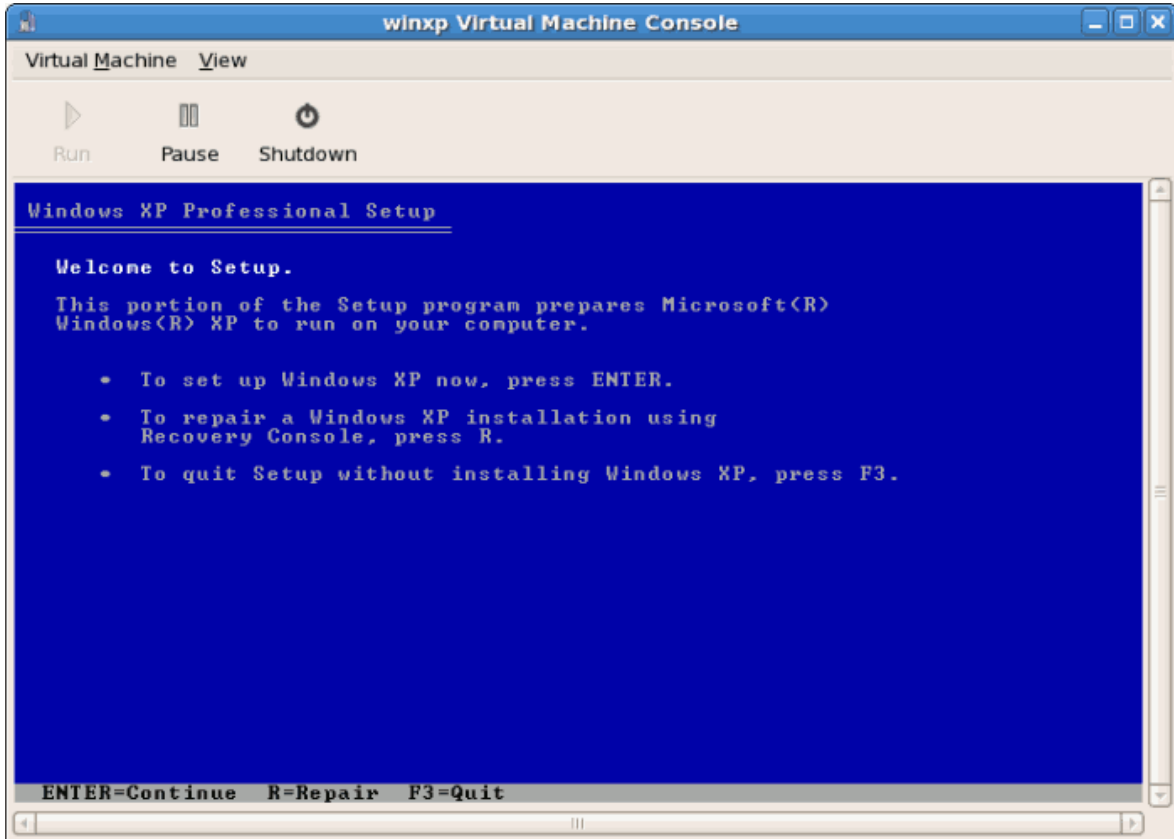
8. 在继续安装前您会看到小结页面。按 完成 按钮执行安装：



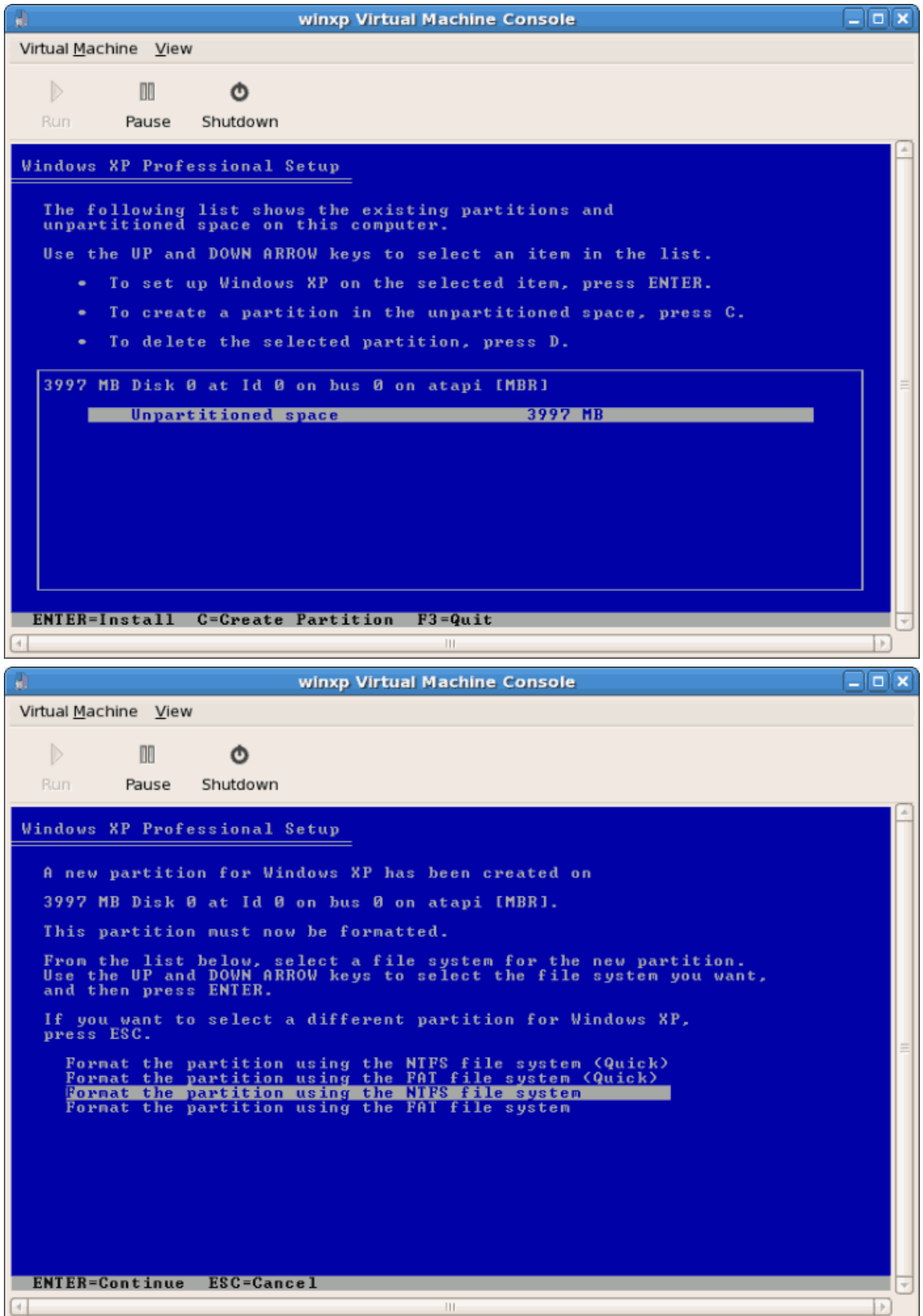
9. 您必须选择硬件，因此请在安装开始后迅速打开一个控制台窗口。点击 完成 按钮后切换到 virt-manager 概述窗口，并选择您新启动的 Windows 客户端。双击系统名称则会打开控制台窗口。迅速重复按 F5 键来选择新的 HAL，当您在 Windows 安装中看到对话框时，请选择 '普通 i486 平台' 标签（您可以使用 Up 和 Down 箭头键在选项中进行滚动选择）。



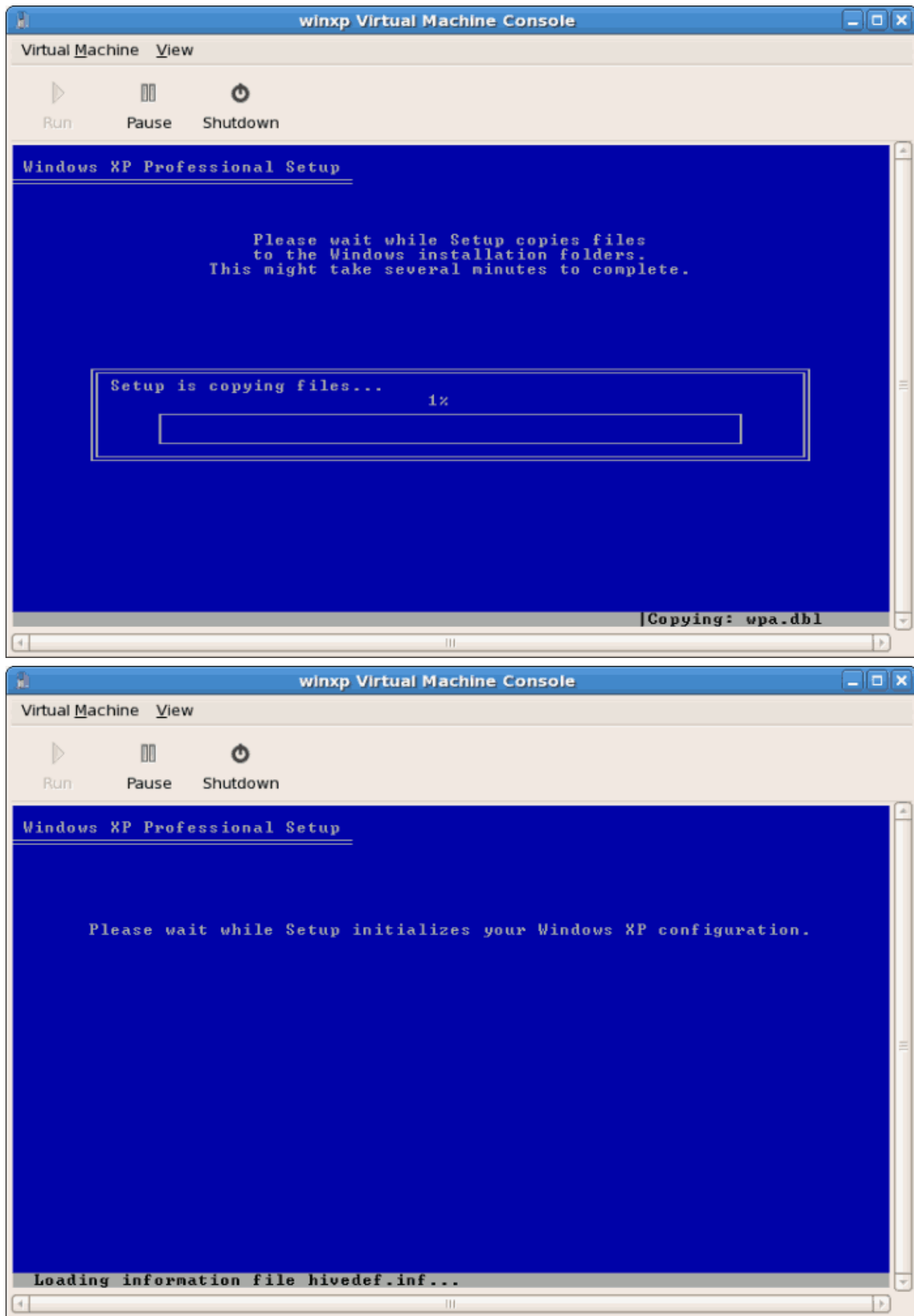
10. 使用标准 Windows 安装继续安装。



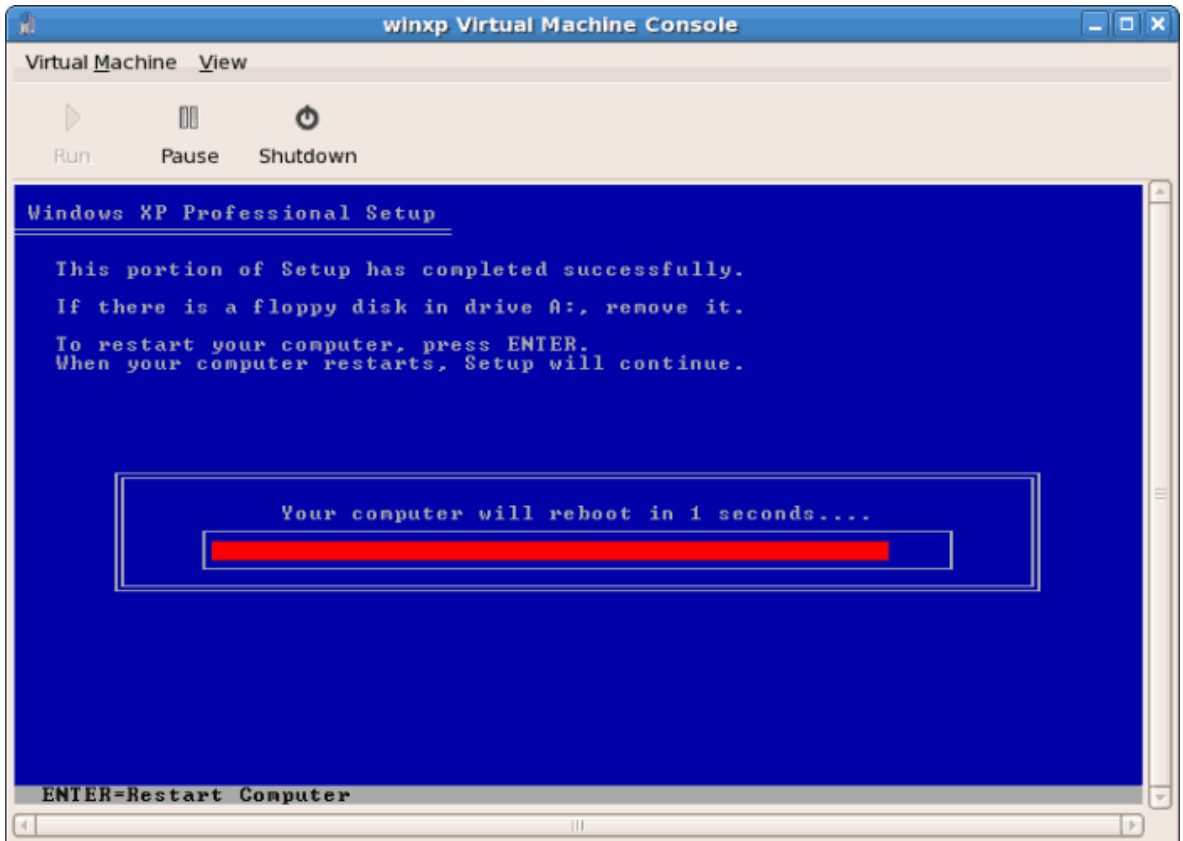
11. 看到提示后对硬盘进行分区。



12. 格式化驱动器后，Windows 会开始将文件复制到硬盘中。



13. 将文件复制到存储设备中，现在重启 Windows。

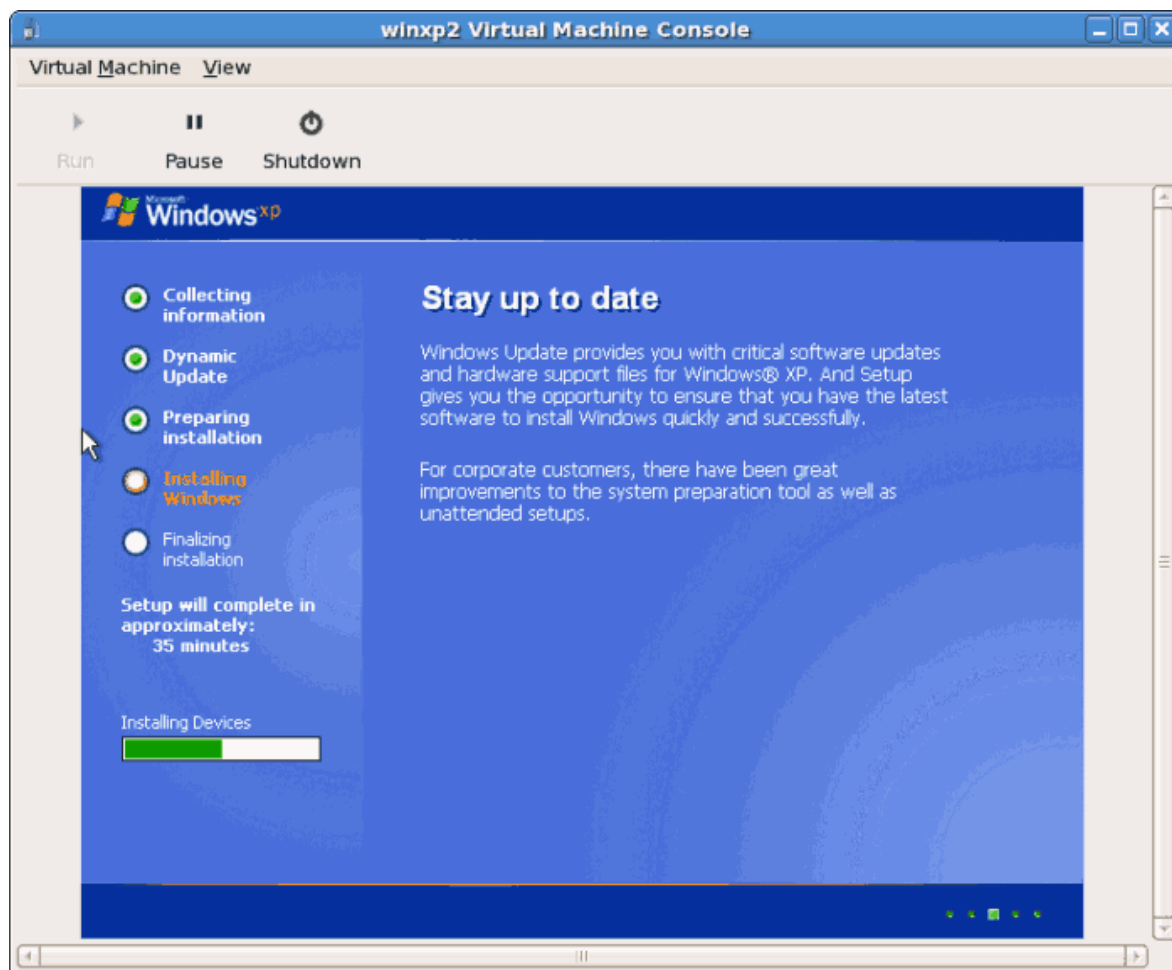


14. 重启您的 Windows 客户端：

```
# virsh start WindowsGuest
```

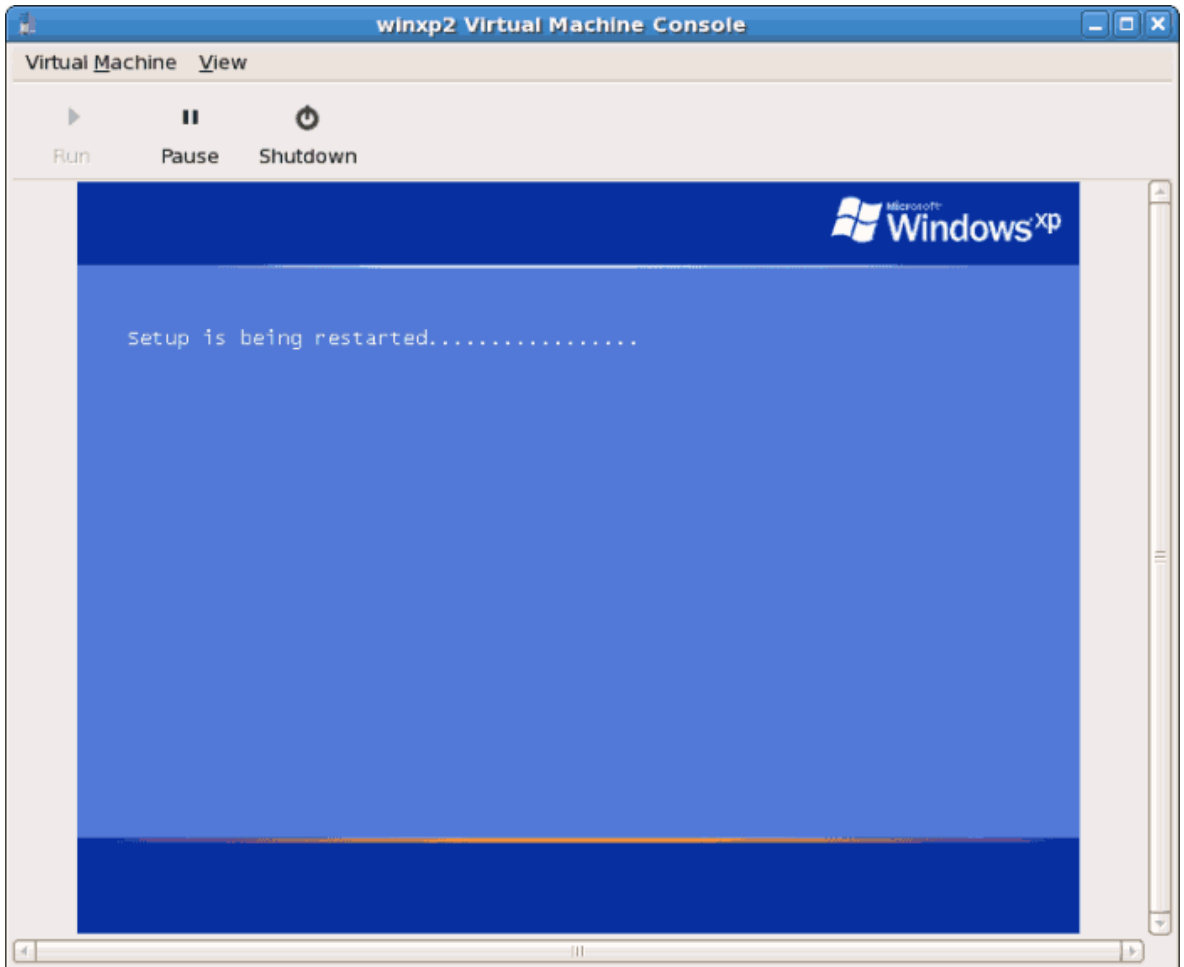
其中 *WindowsGuest* 是您的虚拟机名称。

15. 打开控制台窗口时，您将看到 Windows 安装的设置阶段。

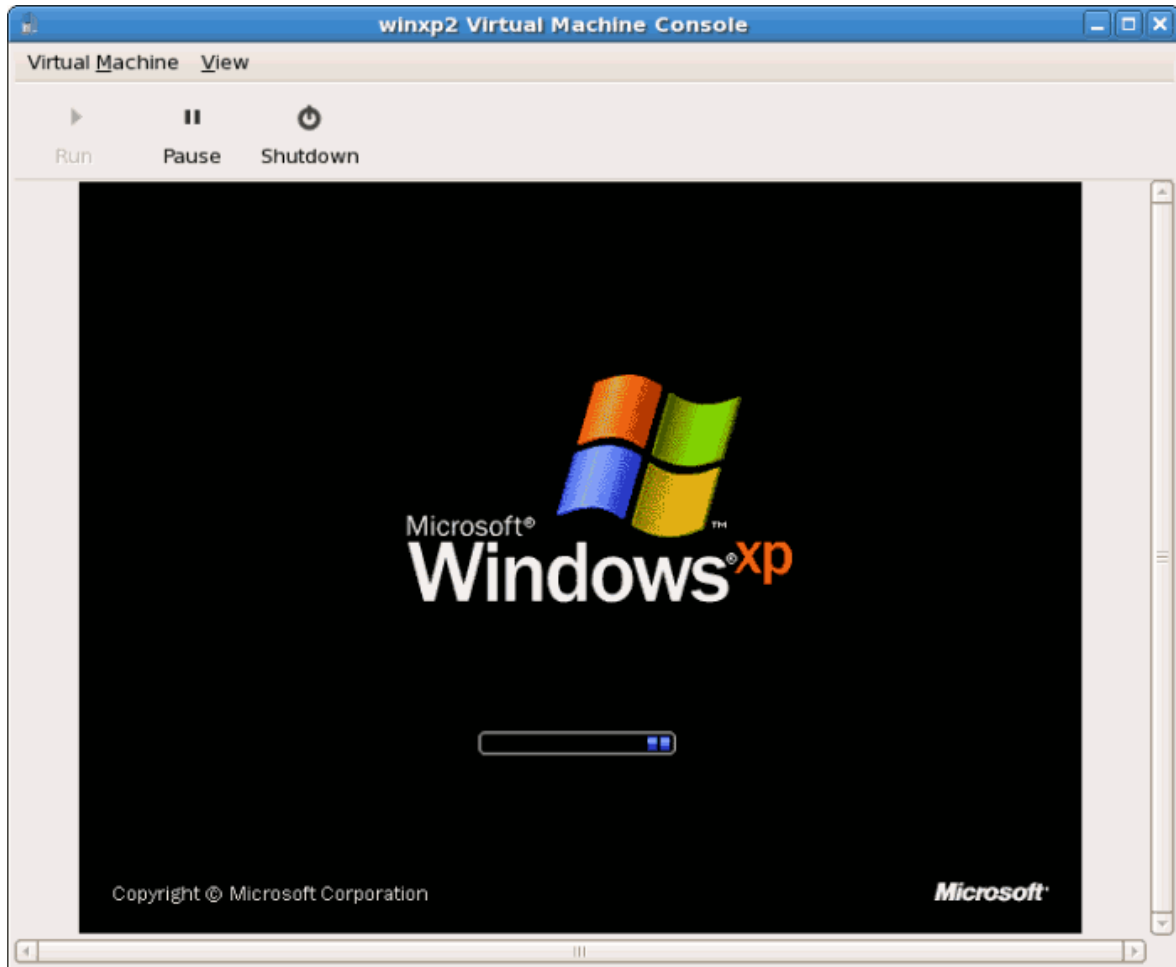


16. 如果您的安装看起来停滞在设置阶段，您可以使用 `virsh reboot WindowsGuestName` 重启该客户端。这样通常可使得安装继续进行。您重启虚拟机后会看到如下信息：Setup is being restarted。

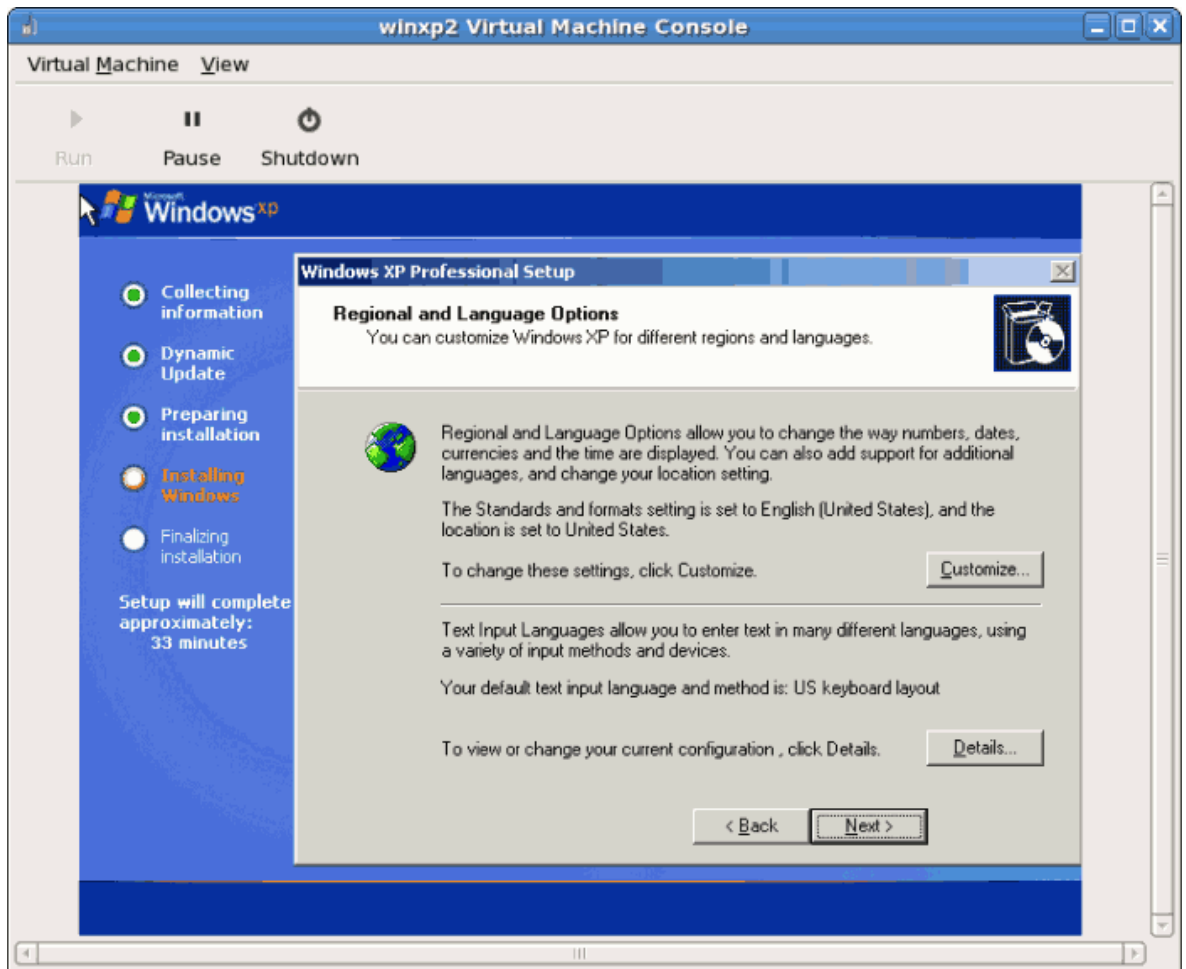




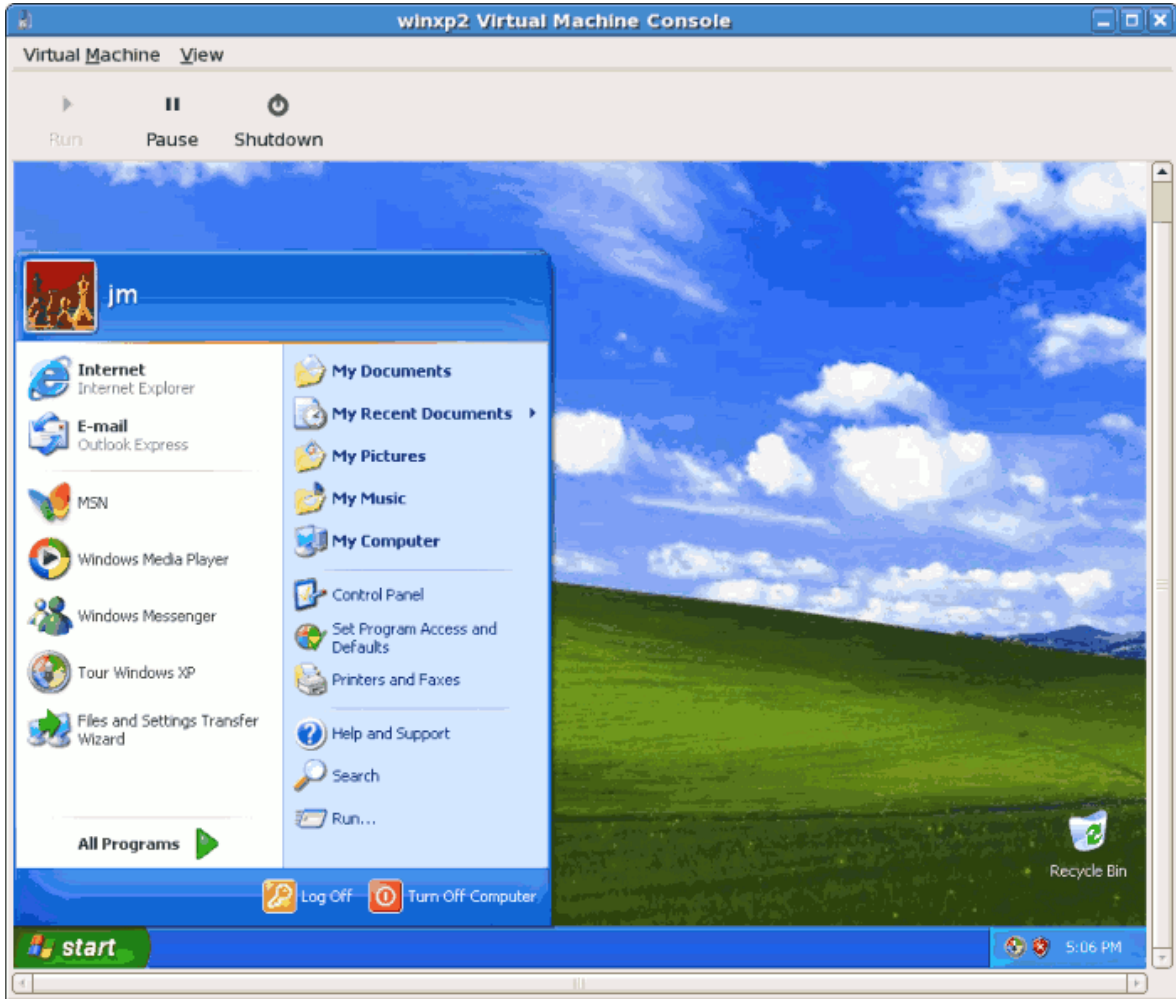
17. 完成设定后，您会看到 Windows 引导屏幕：



18. 您现在可以继续对您的 Windows 安装进行标准设定：



19. 设置过程已完成，出现 Windows 桌面。



### 3.4. 将 Windows 服务器 2003 作为全虚拟客户端安装

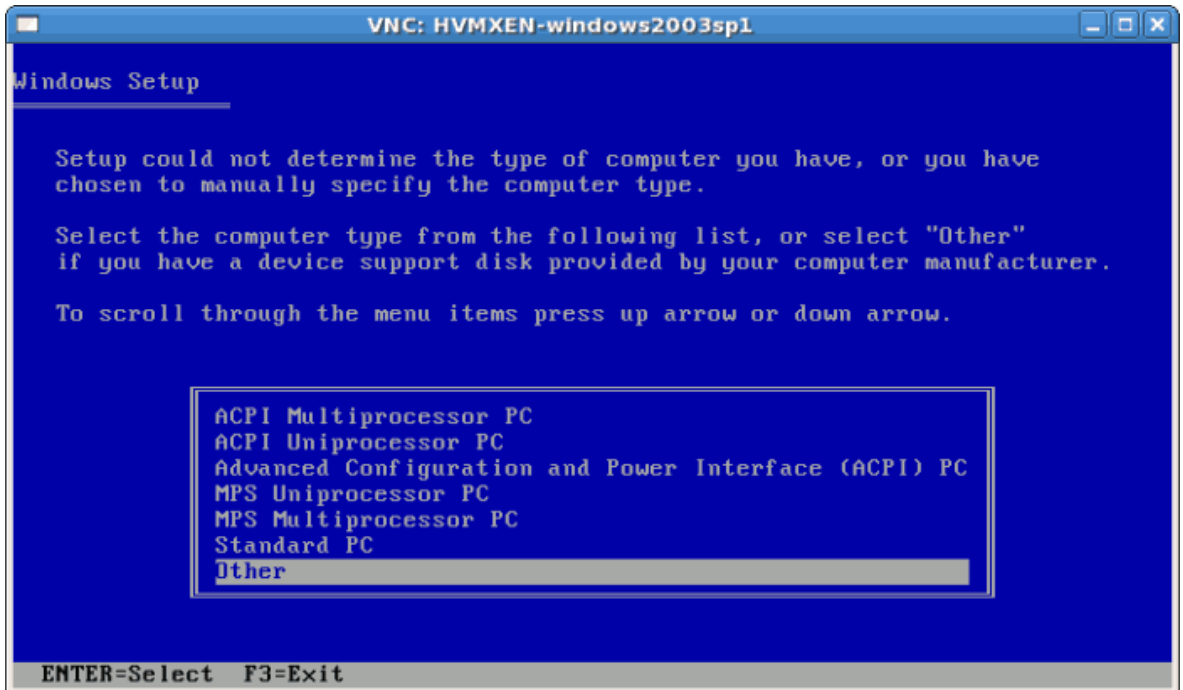
本章论述了使用 virt-install 命令安装全虚拟 Windows 服务器 2003 客户端。可使用 virt-install 替换 virt-manager。这个过程与 3.3 节“Windows XP 全虚拟安装”中所述 Windows XP 安装类似。

1. 使用 virt-install 安装 Windows 服务器 2003，将其作为 Windows 客户端控制台迅速打开 virt-viewer 窗口。使用 virt-install 安装 Windows 服务器 2003 客户端示例如下：

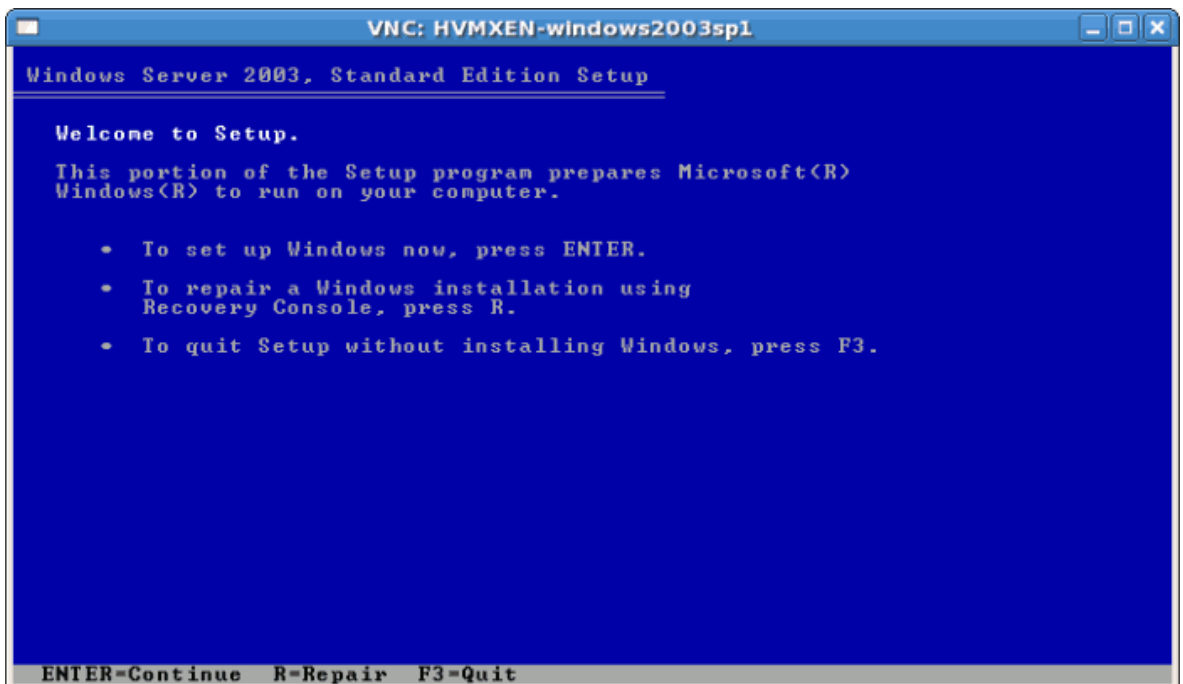
使用 virt-install 命令启动安装。

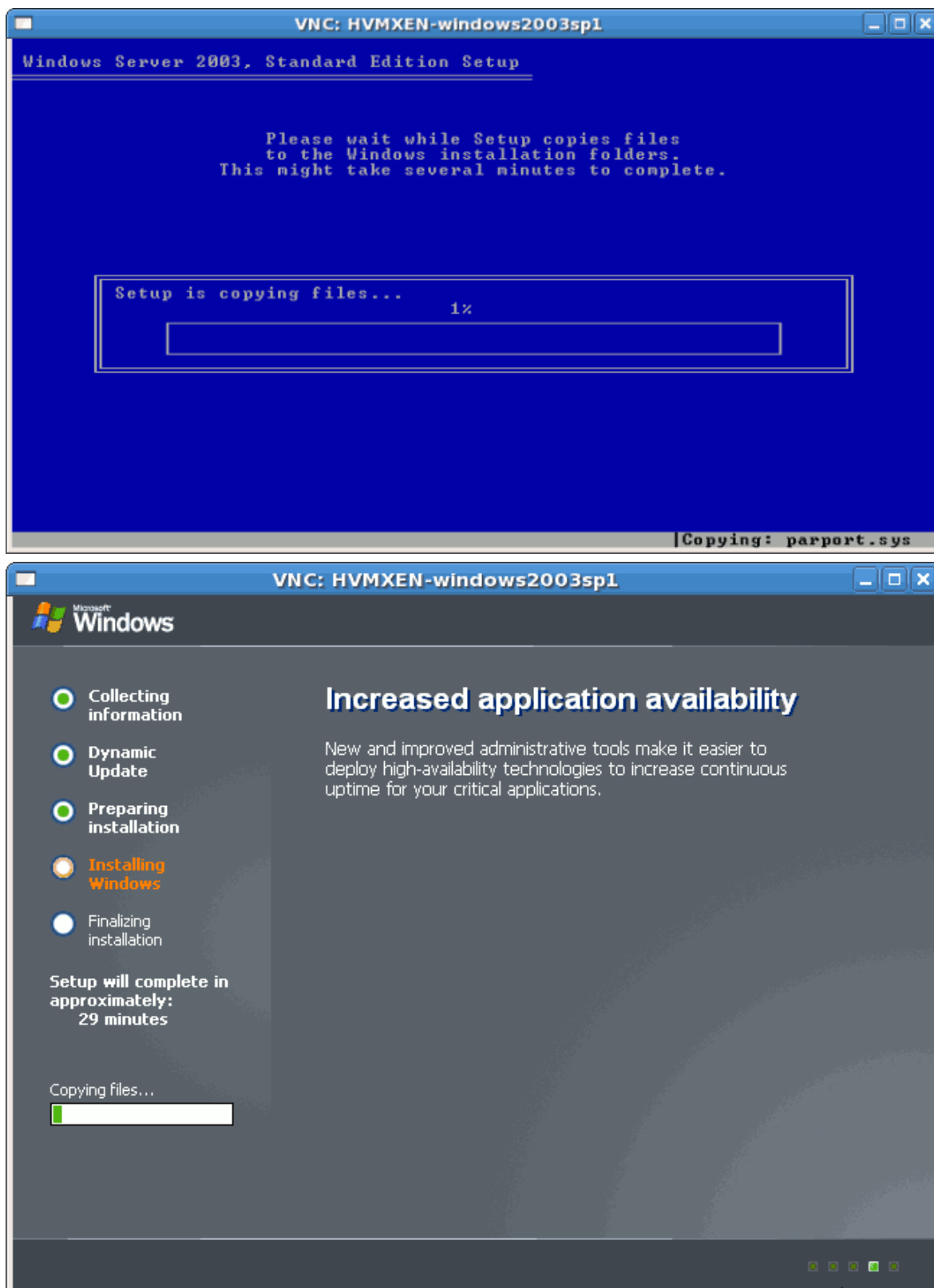
```
# virt-install -hvm -s 5 -f /var/lib/libvirt/images/windows2003sp1.dsk \
-n windows2003sp1 -cdrom=/ISOs/WIN/en_windows_server_2003_sp1.iso \
-vnc -r 1024
```

2. 客户端引导到安装程序后，您需要迅速按下 F5 键。如果您没有及时按下 F5 键，则您需要重新启动安装程序。按 F5 可让您选择不同的「AL」或者「计算机类型」。选择 Standard PC 作为「计算机类型」。在 Windows 服务器 2003 虚拟客户端中需要更改「计算机类型」。



3. 完成余下的安装。





4. 作为全虚拟客户端安装 Windows 2003。

## 3.5. 将 Windows 服务器 2008 作为全虚拟客户端安装

这部分论述如何安装全虚拟 Windows 服务器 2008 客户端。

过程 3.4. 使用 virt-manager 安装 Windows 服务器 2008

1. Open virt-manager  
Start virt-manager. Launch the Virtual Machine Manager application from the Applications menu and System Tools submenu. Alternatively, run the virt-manager command as root.
2. Select the hypervisor  
Select the hypervisor. If installed, select Xen or KVM. For this example, select KVM. Note that presently KVM is named qemu.  
  
选择该选项后「新建」按钮就变得可用了。按「新建」按钮。
3. Start the new virtual machine wizard  
Pressing the New button starts the virtual machine creation wizard.



Press Forward to continue.

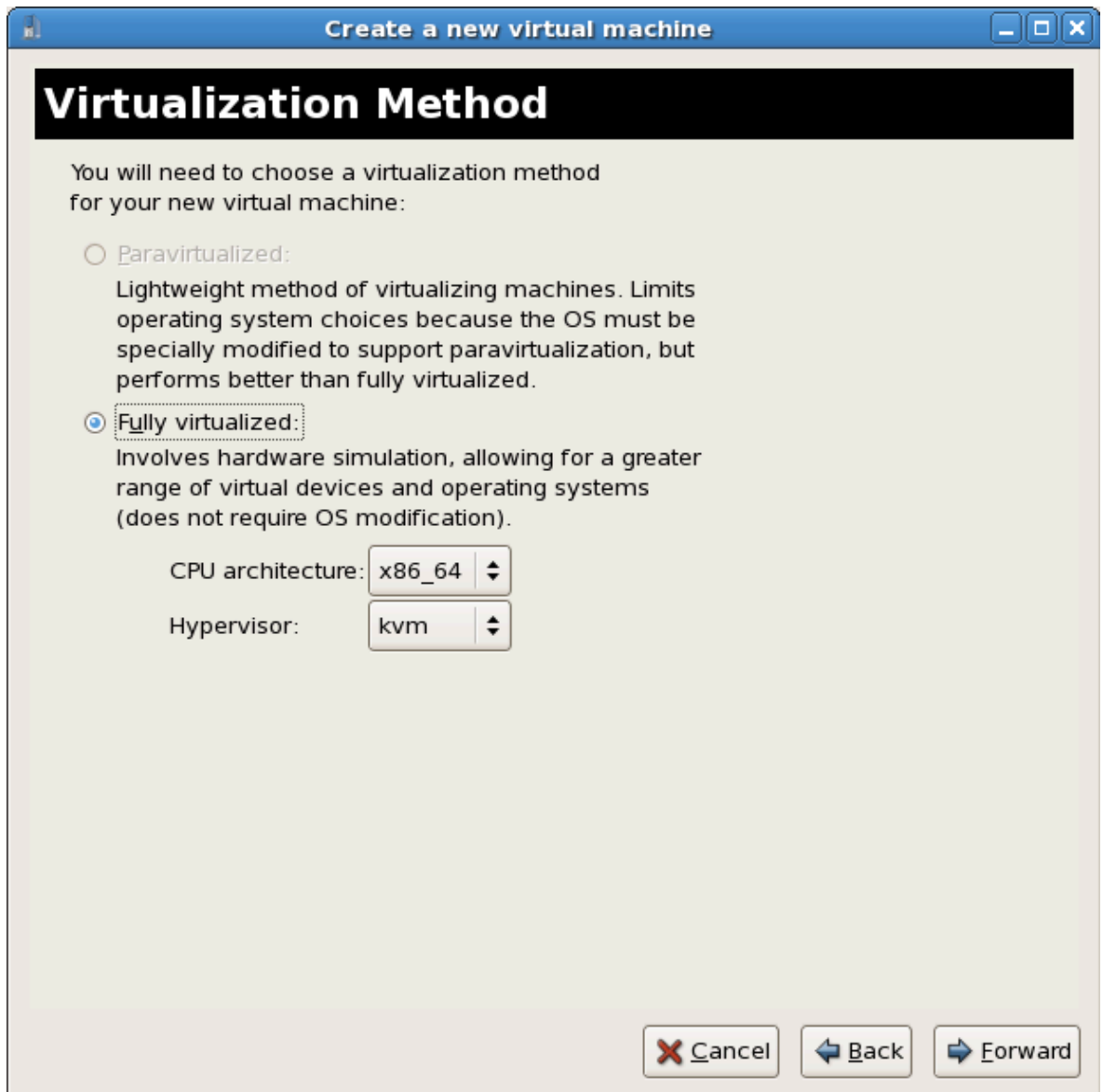
- 4. Name the virtual machine  
为虚拟机客户端提供名称。不允许有标点和空白字符。



按 前进 继续。

- 5. Choose a virtualization method  
为虚拟客户端选择虚拟化方法。请注意：您只能选择安装了虚拟化方法。如果您之前选择了 KVM 或者 Xen（第二步），您必须使用您选择的管理程序。这个示例使用 KVM 管理程序。





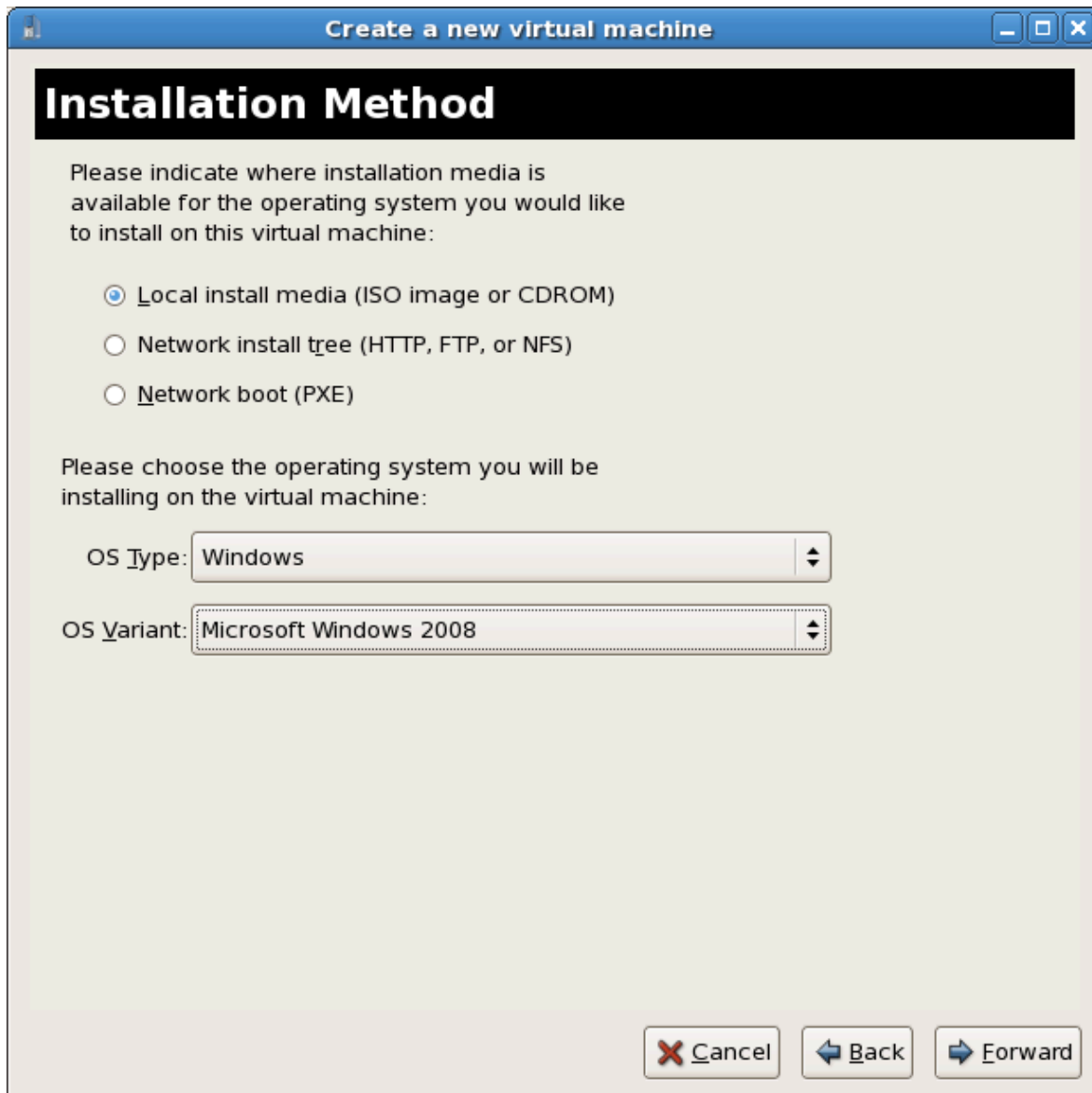
按 前进 继续。

6. Select the installation method

对于所有 Windows 版本您必须使用「本地安装介质」，可以是 ISO 映像或者物理光学介质。

如果您为 Windows 网络安装配置了 PXE 服务器，则您可以使用 PXE。PXE Windows 安装不在本章讨论范围内。

将「操作系统」类型设定为「Windows」；「操作系统变体」设定为「微软 Windows 2008」，如截屏中所示。

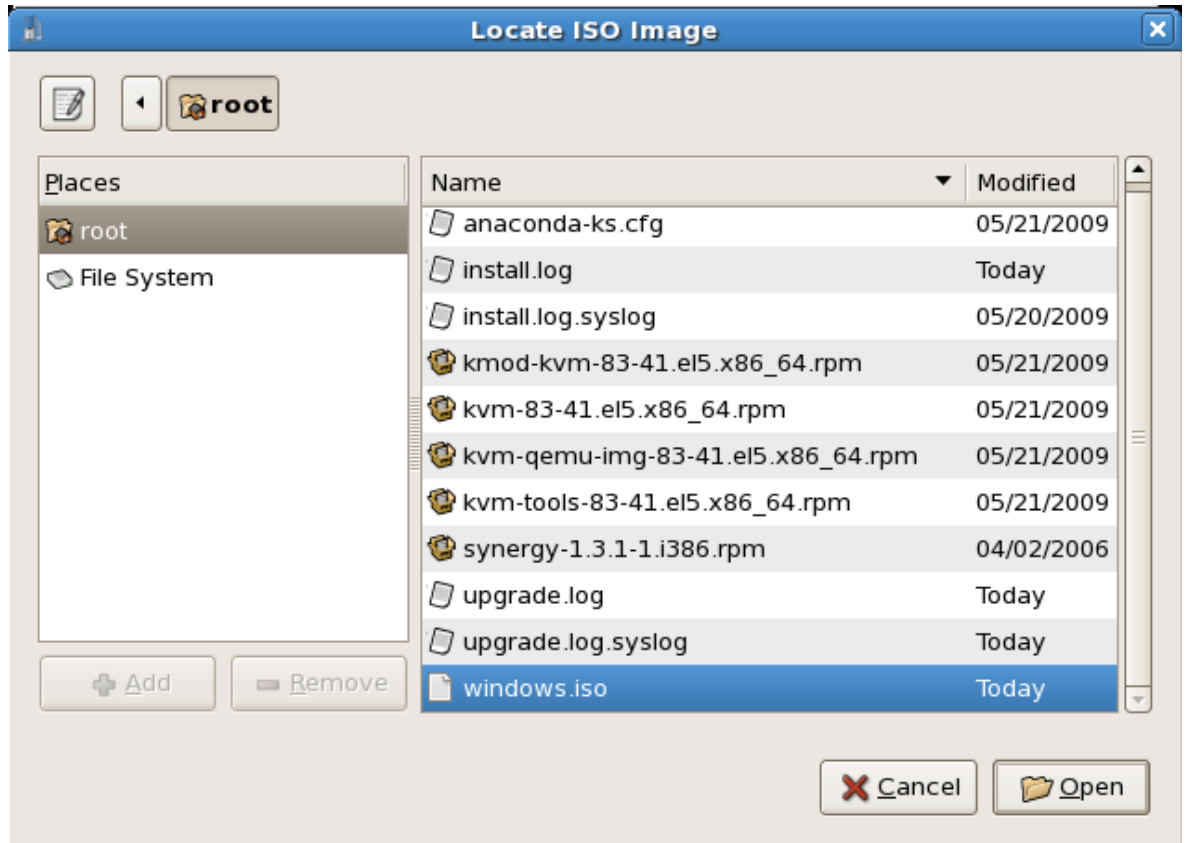


按 前进 继续。

7. Locate installation media

选择 ISO 映像位置或者 CD-ROM 或者 DVD 设备。这个示例使用 Windows 服务器 2008 安装磁盘的 ISO 文件映像。

- a. Press the Browse button.
- b. 搜索 ISO 文件位置并选择它。



按 打开 确认您的选择。

- c. 选择该文件并准备使用其进行安装。



按 前进 继续。



### Image files and SELinux

推荐使用的 ISO 映像文件和存储映像在 `/var/lib/libvirt/images/` 目录中。其它位置可能需要对 SELinux 进行额外配置，详情请参考 7.1 “SELinux 配置”。

#### 8. Storage setup

分配一个物理存储设备（「块设备」）或者基于文件的映像（「文件」）。基于文件的映像必须保存在 `/var/lib/libvirt/images/` 目录中。请为您的虚拟客户端及其所需程序分配足够空间。



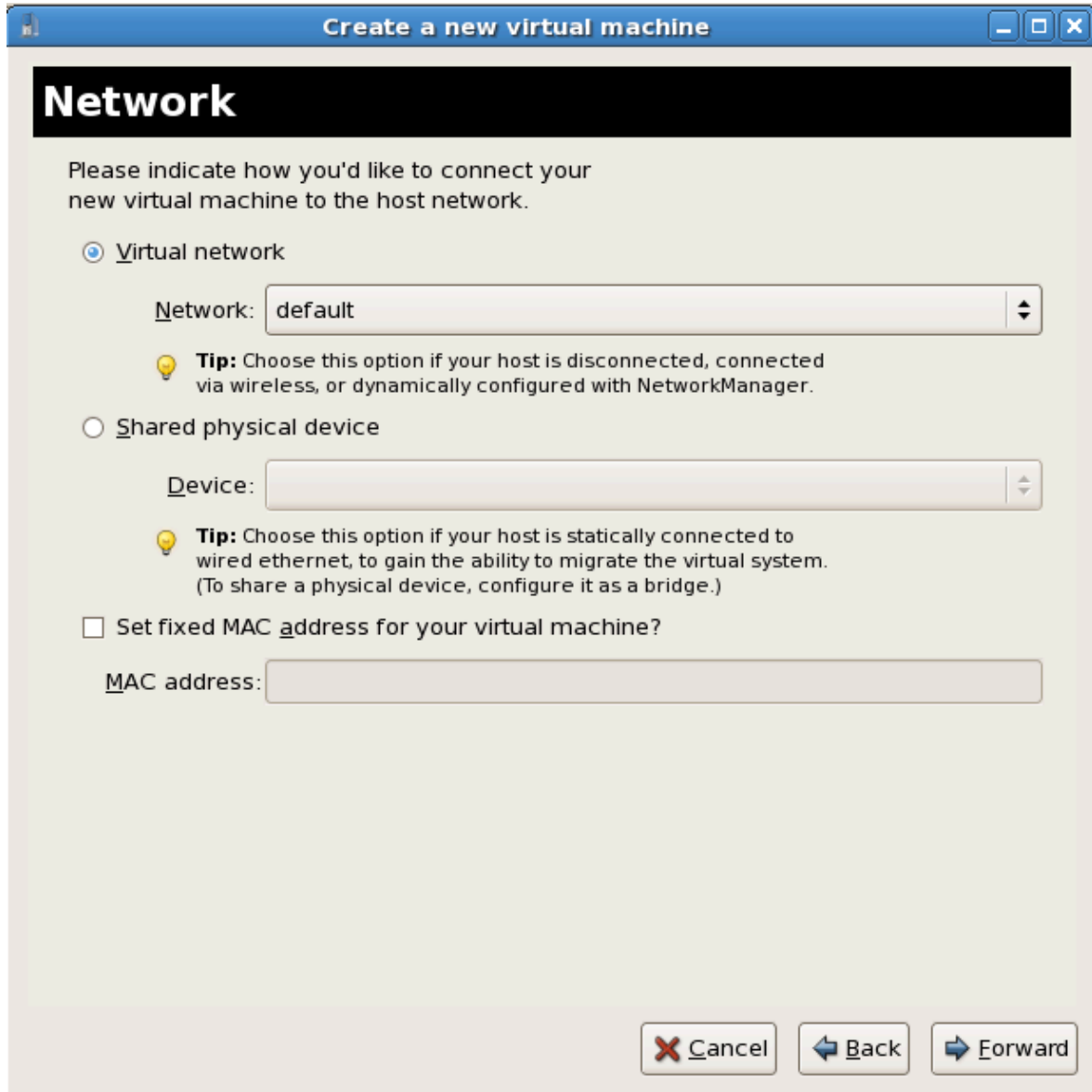
按 前进 继续。

9. Network setup

Select either Virtual network or Shared physical device.

The virtual network option uses Network Address Translation (NAT) to share the default network device with the virtualized guest. Use the virtual network option for wireless networks.

The shared physical device option uses a network bond to give the virtualized guest full access to a network device.



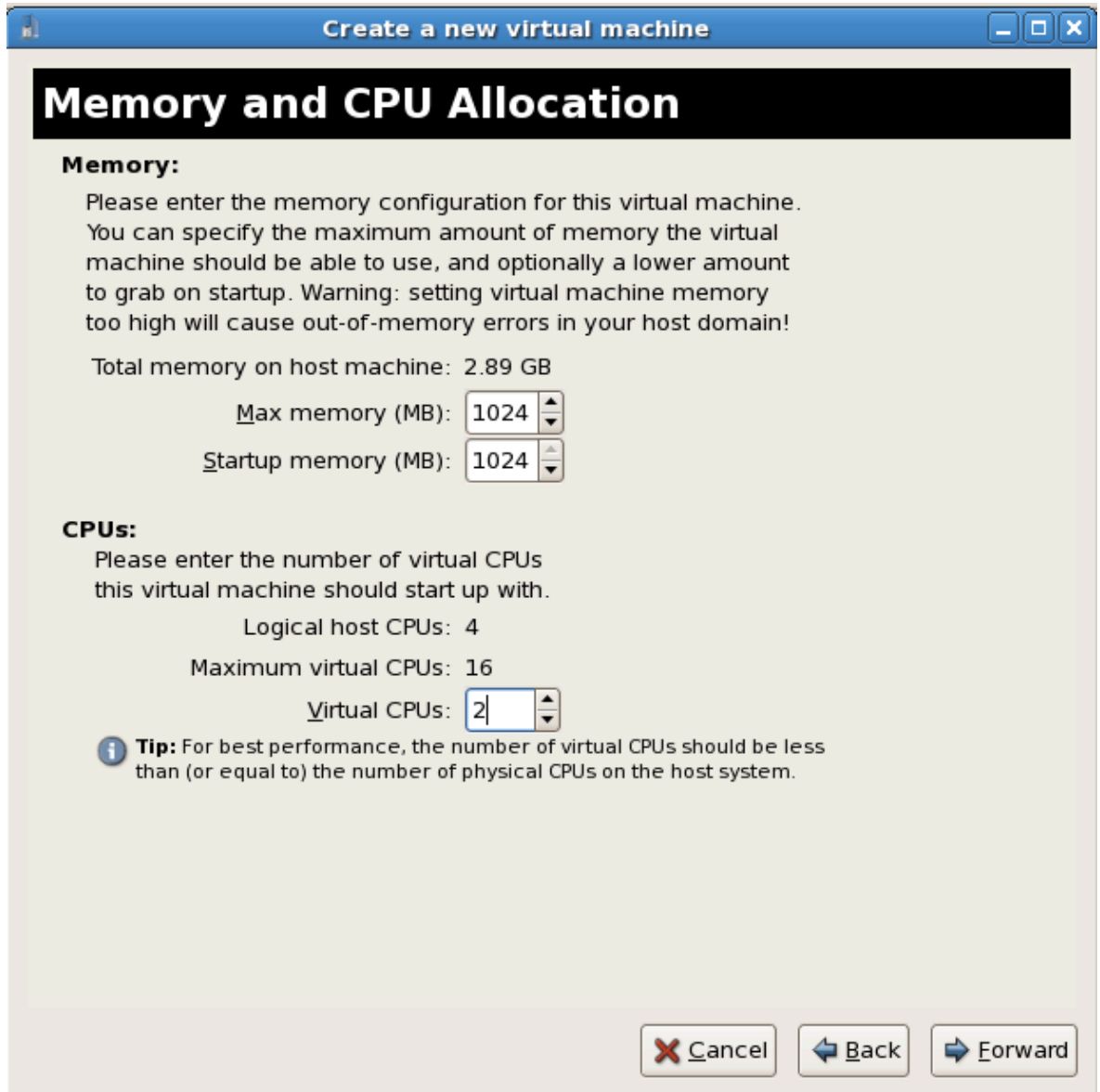
Press Forward to continue.

#### 10. Memory and CPU allocation

The Allocate memory and CPU window displays. Choose appropriate values for the virtualized CPUs and RAM allocation. These values affect the host's and guest's performance.

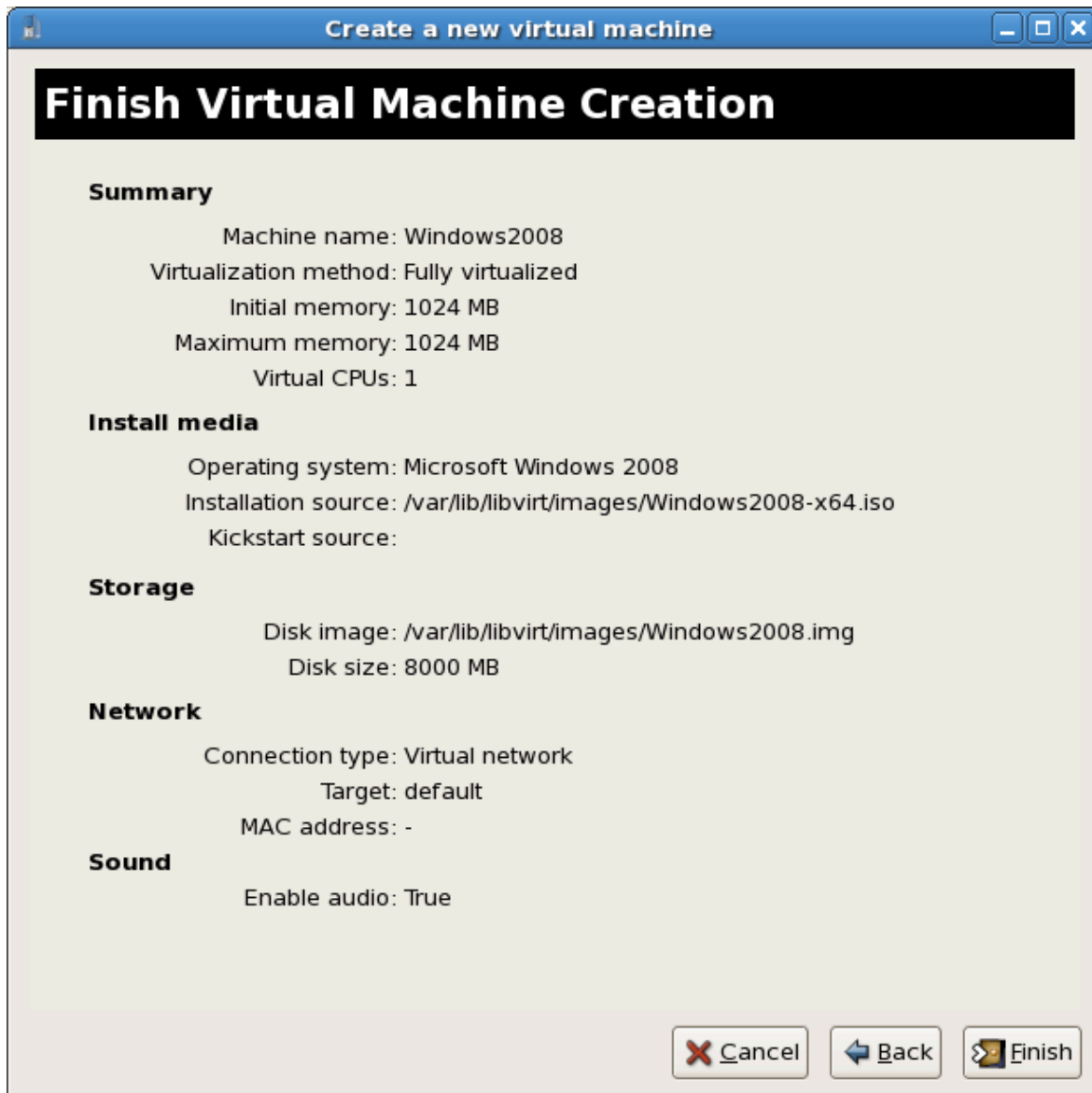
Virtualized guests require sufficient physical memory (RAM) to run efficiently and effectively. Choose a memory value which suits your guest operating system and application requirements. Windows Server 2008. Remember, guests use physical RAM. Running too many guests or leaving insufficient memory for the host system results in significant usage of virtual memory and swapping. Virtual memory is significantly slower causing degraded system performance and responsiveness. Ensure to allocate sufficient memory for all guests and the host to operate effectively.

Assign sufficient virtual CPUs for the virtualized guest. If the guest runs a multithreaded application assign the number of virtualized CPUs it requires to run most efficiently. Do not assign more virtual CPUs than there are physical processors (or hyper-threads) available on the host system. It is possible to over allocate virtual processors, however, over allocating has a significant, negative affect on guest and host performance due to processor context switching overheads.



Press Forward to continue.

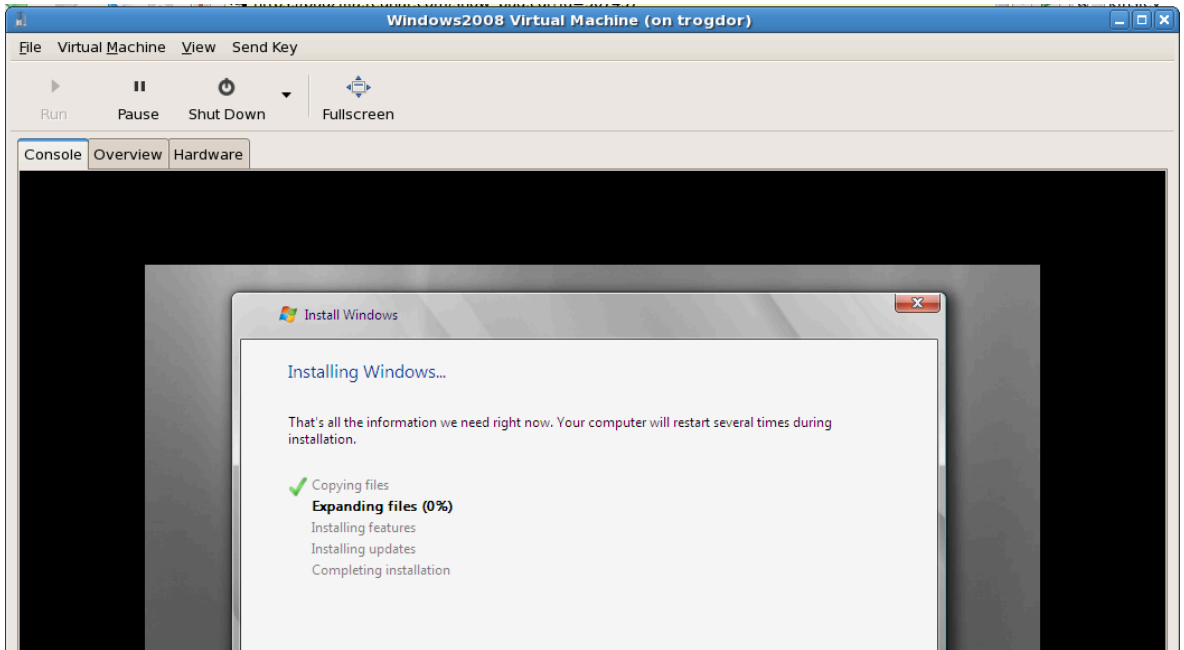
11. Verify and start guest installation  
验证配置。



按 完成 启动客户端安装过程。



## 12. 安装 Windows



完成 Windows 服务器 2008 安装程序。本指南不包括安装程序，有关安装 Windows 的信息请参考微软的 [documentation](#)<sup>1</sup>。



---

## 部分 II. Configuration

### 在 Fedora 这配置虚拟化

这些章节包括各种高级虚拟化任务配置步骤。这些任务包括添加网络和存储设备、提高的安全性、改进的性能以及在全虚拟客户端中使用半虚拟驱动程序。

---

---

## 虚拟块设备

本章讲述如何在红帽虚拟化客户端中安装和配置块设备。名词块设备指的是各种形式的存储设备。

### 4.1.0 创建一个虚拟软盘控制程序

软盘控制程序需要很多老的操作系统，特别是在安装驱动程序时。目前无法从虚拟客户端访问物理软盘设备，但支持从虚拟软盘设备创建并访问软盘磁盘映像。这部分论述的是如何创建虚拟软盘设备。

需要一个软盘映像文件。请使用 `dd` 命令创建软盘映像文件。使用正确的软盘设备名称及磁盘名称替换 `/dev/fd0`。

```
# dd if=/dev/fd0 of=~/.legacydrivers.img
```



#### 半虚拟驱动程序备注

半虚拟驱动程序可将物理软盘设备与全虚拟客户端对应。

这个示例使用 `virt-manager` 创建的客户端，该客户端使用位于 `/var/lib/libvirt/images/rhel5FV.img` 的映像运行全虚拟红帽企业版 Linux 安装。在这个示例中使用 `Xen` 管理程序。

1. 请在运行的客户端中使用 `virsh` 命令为您的客户端映像生成 XML 配置文件。

```
# virsh dumpxml rhel5FV > rhel5FV.xml
```

这样可将配置设定保存为 XML 文件，编辑该文件可优化客户端使用的操作及设备。有关使用 `virsh` XML 配置文件的详情请参考 [18](#) `libvirt`。

2. 为客户端创建软盘映像。

```
# dd if=/dev/zero of=/var/lib/libvirt/images/rhel5FV-floppy.img bs=512 count=2880
```

3. 添加以下内容并为您的客户端配置 XML 文件做相应的修改。这个示例创建了作为基于文件虚拟设备的带软驱的客户端。

```
<disk type='file' device='floppy'>
  <source file='/var/lib/libvirt/images/rhel5FV-floppy.img' />
  <target dev='fda' />
</disk>
```

4. 停止客户端。

```
# virsh stop rhel5FV
```

5. 使用 XML 配置文件重启客户端。

```
# virsh create rhel5FV.xml
```

现在软盘设备在客户端中可用，并作为映像文件保存在主机中。

## 4.2. 在客户端中添加存储设备

这部分论述了如何在虚拟客户端中添加存储设备。只能在创建客户端之后才可添加附加存储。支持的存储设备和协议包括：

- 本地硬盘分区，
- 逻辑卷，
- 光纤通道或者直接连接到主机的 iSCSI。
- 位于主机文件系统中的文件容器。
- 由虚拟机直接挂载的 NFS 文件系统。
- 客户端可直接访问的 iSCSI 存储。
- 群集文件系统（GFS）。

### 在客户端中添加基于文件的存储

基于文件的存储或者基于文件的容器是指主机文件系统中作为虚拟客户端虚拟硬盘使用的文件。要添加基于文件的容器，请执行以下步骤：

1. 创建空白容器文件或者使用现有文件容器（比如 ISO 文件）。
  - a. 请使用 `dd` 命令创建稀疏文件。因为有数据完整性和性能问题，所以不建议使用稀疏文件。创建稀疏文件的速度较快，并可用来进行测试，但不应在产品环境中使用。

```
# dd if=/dev/zero of=/var/lib/libvirt/images/FileName.img bs=1M seek=4096 count=0
```

- b. 建议在基于文件的存储容器中使用非稀疏、预分配的文件。要生成非稀疏文件，请执行：

```
# dd if=/dev/zero of=/var/lib/libvirt/images/FileName.img bs=1M count=4096
```

这两个命令都会生成一个在虚拟客户端中作为附加存储使用的 400MB 大小的文件。

2. 转储客户端配置。在这个示例中，客户端为 *Guest1*，该文件保存在用户主目录中。

```
# virsh dumpxml Guest1 > ~/Guest1.xml
```

3. 请在文本编辑器中打开配置文件（在这个示例中是 *Guest1.xml*）。找到以 “disk=” 开始的条目。这个条目类似如下：

```
>disk type='file' device='disk'<
  >driver name='tap' type='aio'</>
  >source file='/var/lib/libvirt/images/Guest1.img'</>
  >target dev='xvda'</>
>/disk<
```

4. 修改 `disk=` 最后的条目来添加附加存储。确定为虚拟块设备指定了设备名称，且该名称还未在配置文件中使用。以下是添加名为 `FileName.img` 的文件作为基于文件的存储容器的条目示例：

```
>disk type='file' device='disk'<
  >driver name='tap' type='aio'<<
  >source file='/var/lib/libvirt/images/Guest1.img'<<
  >target dev='xvda'<<
>/disk<
>disk type='file' device='disk'<
  >driver name='tap' type='aio'<<
  >source file='/var/lib/libvirt/images/FileName.img'<<
  >target dev='hda'<<
>/disk<
```

5. 使用更新的配置文件重新启动客户端。

```
# virsh create Guest1.xml
```

6. 以下步骤特别用于 Linux 客户端。其它操作系统使用不同方法处理新存储设备。对于非 Linux 系统，请参考您客户端操作系统文档。

该客户端现在使用文件 `FileName.img` 作为设备 `/dev/hdb`。这个设备需要在该客户端中进行格式化。在该客户端中，将整个设备分为一个主分区，然后格式化该设备。

- a. 按 *n* 添加新分区。

```
# fdisk /dev/hdb
Command (m for help):
```

- b. 按 *p* 添加主分区。

```
Command action
 e  extended
 p  primary partition (1-4)
```

- c. 选择可用分区号。在这个示例中，输入 **1** 选择第一个分区。

```
分区号 (1-4) : 1
```

- d. 按 *Enter* 输入默认首柱面。

```
首柱面 (1-400, 默认为 1) :
```

- e. 选择分区大小。在这个示例中，按 *Enter* 则分配整个磁盘。

```
尾柱面或者 +size、+sizeM、+sizeK (2-400, 默认为 400) :
```

- f. 按 *t* 选择分区类型。

```
命令 (m 为帮助) : t
```

- g. 选择在上一步创建的分区。在这个示例中为分区 **1**。

```
分区号 (1-4) : 1
```

- h. 输入 **83** 选择 Linux 分区。

```
Hex 代码 (输入 L 列出代码) : 83
```

- i. 在磁盘中写入更改并退出。

```
命令 (m 为帮助) : w
```

```
命令 (m 为帮助) : q
```

- j. 用 ext3 文件系统格式化新分区。

```
# mke2fs -j /dev/hdb
```

7. 在客户端中挂载该磁盘。

```
# mount /dev/hdb1 /myfiles
```

该客户端现在拥有一个基于文件的虚拟存储设备。

### 在客户端中添加硬盘及其它块设备

系统管理员可使用附加硬盘提供更多存储空间或者从用户数据中分离系统数据。这个步骤，即 [00 4.1, “00 000000000000”](#)，描述了如何将主机中的硬盘添加到虚拟客户端中。

这个步骤适用于所有物理块设备，其中包括 CD-ROM、DVD 及软盘设备。

#### 过程04.1.0在虚拟客户端中添加物理块设备

1. 在该主机中添加物理硬盘。如果无法默认访问该驱动器，请配置该主机。
2. 请使用 `multipath` 配置该设备，如果需要则在该主机中永久保留。
3. 请使用 `virsh attach` 命令。使用您的客户端名称替换 `myguest`，使用要添加的设备替换 `/dev/hdb1`，使用客户端中该设备的位置替换 `hdc`。`hdc` 必须是一个未使用的设备名称。在 Windows 客户端中也使用 `hd*` 标记，客户端可正确识别该设备。

在该命令中为 CD-ROM 或者 DVD 设备附加 `--type hdd` 参数。

在该命令中为软盘设备附加 `--type floppy` 参数。

```
# virsh attach-disk myguest /dev/hdb1 hdc --driver tap --mode readonly
```



4. 该客户端现在有一个新硬盘设备，在 Linux 中名为 `/dev/hdb`，在 Windows 中名为 `D: drive`。这个设备可能需要格式化。

### 4.3.0 配置持久存储

这部分论述的是使用外部或者联网存储的系统，即光纤通道或者基于 iSCSI 的存储设备。建议在您的主机中为那些系统配置持久设备名称。这还可已帮助实时迁移并为多个虚拟系统提供一致的设备名称和存储。

全局唯一标识符（Universally Unique Identifier，UUID）是在分散的计算环境中识别计算机和设备的方法。这部分介绍了使用 UUID 来识别 iSCSI 或者光纤通道 LUN。重启、断开连接或者设备交换后 UUID 不会改变。UUID 类似于设备的标签。

没有运行 `multipath` 的系统必须使用 `options=-b`。运行 `multipath` 的系统可以使用 `options=-g`。

#### 单一路径配置

这个步骤使用 `udev` 持久部署 `LUN` 设备。只能在不使用 `multipath` 的主机中使用这个步骤。

1. 编辑 `/etc/scsi_id.config` 文件。
  - a. 确定注释出 `options=-b` 行。

```
# options=-b
```

- b. 请添加以下行：

```
options=-g
```

这个选项会将 `udev` 配置为假设所有附加的 SCSI 都会返回 UUID。

2. 要显示给定设备的 UUID，请执行以下命令：`scsi_id -g -s /block/sd*`。例如：

```
# scsi_id -g -s /block/sd*
3600a0b800013275100000015427b625e
```

输出的结果可能与以上示例有所不同。输出会显示设备 `/dev/sdc` 的 UUID。

3. 确认 `scsi_id -g -s /block/sd*` 命令输出的 UUID 与访问该设备的计算机中的一致。
4. 为该设备名称创建规则。请在 `/etc/udev/rules.d` 目录中创建名为 `20-names.rules` 的文件。添加到同一文件中的所有规则都使用相同的格式。规则格式如下：

```
KERNEL="sd*",BUS="scsi",PROGRAM="/sbin/scsi_id -g -s",RESULT=UUID,NAME=devicename
```

请使用上面检索的 UUID 和设备名称替换 `UUID` 和 `devicename`。这是如上示例中的一个规则：

```
KERNEL="sd*",BUS="scsi",PROGRAM="/sbin/scsi_id -g -s",
RESULT="3600a0b800013275100000015427b625e",NAME="rack4row16"
```

udev 守护进程现在为该规则中的 UUID 搜索所有名为 /dev/sd\* 的设备。当将匹配的设备连接到系统后，会根据该规则为这个设备分配一个名称。UUID 为 3600a0b800013275100000015427b625e 的设备会以 /dev/rack4row16 的形式出现。

- 在 /etc/rc.local 中附加这一行：

```
/sbin/start_udev
```

- 将 /etc/scsi\_id.config、/etc/udev/rules.d/20-names.rules 和 /etc/rc.local 文件中的更改复制到所有相关主机中。

```
/sbin/start_udev
```

在更新这些文件的主机中，使用配置规则的联网存储设备现在都有持久的名称。就是说您可以在使用共享存储的主机间迁移客户端，且那些客户端可以访问其配置文件中包含的存储设备。

### 多路径配置

multipath 软件包用于在计算机和存储设备间有超过一个物理路径的系统。multipath 为附加到 Linux 系统的网络存储设备提供容错、故障切换以及改进的性能。

要在 multipath 环境中部署 LUN 持久性，需要为您的多路径设备定义别名。每个存储设备都有一个 UUID，它是作为别名的密钥使用。请使用 scsi\_id 命令识别设备的 UUID。

```
# scsi_id -g -s /block/sdc
```

多路径设备将在 /dev/mpath 目录中生成。在下面的示例中 /etc/multipath.conf 文件定义了 4 个设备：

```

multipaths {
  multipath {
    wwid      3600805f30015987000000000768a0019
    alias     oramp1
  }
  multipath {
    wwid      3600805f30015987000000000d643001a
    alias     oramp2
  }
  mulitpath {
    wwid      3600805f3001598700000000086fc001b
    alias     oramp3
  }
  mulitpath {
    wwid      3600805f300159870000000000984001c
    alias     oramp4
  }
}

```

这个配置将生成 4 个 LUN，分别名为 /dev/mpath/oramp1、/dev/mpath/oramp2、/dev/mpath/oramp3 和 /dev/mpath/oramp4。输入后，设备的 WWID 将与其新名称匹配，并在重启后保留。

## 4.4.0 在客户端中添加虚拟光驱或者 DVD 设备

要在客户端在线时在其中附加一个 ISO 文件，请使用带 *attach-disk* 参数的 *virsh* 命令。

```
# virsh attach-disk [domain-id] [source] [target] --driver file --type cdrom --mode readonly
```

*source* 和 *target* 参数分别是主机和客户端中文件和设备的路径。 *source* 参数可以是到 ISO 文件的路径或者到 */dev* 目录中设备的路径。



## 共享存储和虚拟化

本章论述了在 Fedora 中使用带虚拟化的联网和联网存储。

支持以下虚拟化方法：

- Fibre Channel
- iSCSI
- NFS
- GFS2

联网的存储对实时和离线客户端迁移很重要。您不能在缺少共享存储的情况下迁移客户端。

### 5.1. 使用 iSCSI 存储客户端

这部分论述使用基于 iSCSI 设备存储虚拟客户端。

### 5.2. 使用 NFS 存储客户端

这部分论述使用 NFS 存储虚拟客户端。

### 5.3. 使用 GFS2 存储客户端

这部分论述使用 fedora 全局文件系统 2 (GFS2) 存储虚拟客户端。



## 服务器最佳实践

以下任务和提示可帮助您保证红帽企业版 Fedora 服务器主机（dom0）的可靠性。

- 以强制模式（enforcing）运行 SELinux。您可以执行以下命令来达到此目的。

```
# setenforce 1
```

- 删除或者禁用所有不必要的服务，例如 AutoFS、NFS、FTP、HTTP、NIS、telnetd、sendmail 等等。
- 在服务器中只添加平台管理所需的最小用户帐户数，并删除不必要的用户帐户。
- 避免在您的主机中运行任何不重要的应用程序。在主机中运行应用程序可能会影响虚拟机性能，并可影响服务器稳定性。任何可能使服务器崩溃的应用程序也会导致服务器中的所有虚拟机无法工作。
- 使用中央位置保存虚拟机安装程序及映像。虚拟机映像应该保存在 /var/lib/libvirt/images/ 目录中。如果您要为您的虚拟机使用不同的目录，请确定将该目录添加到您的 SELinux 策略中并在开始安装前对其进行重新标记。
- 应该将安装源、安装树和安装映像保存在中央位置，通常是您的 vsftpd 服务器。





## 虚拟化的安全性

当在企业构架中部署虚拟化技术时，您必须确定不会损害到主机。在 Xen 管理程序中，主机是进行系统管理并管理所有虚拟机的特权域。如果主机不安全，那么该系统中的其它所有域都有危险。有几种方法可使用虚拟化增强安全性。您或者您的机构应该创建一个模板，其内容包括操作说明，并指定在您的虚拟客户端和主机服务器中需要的服务，以及这些服务所需要的支持。以下是您在制定实施计划时应该考虑的安全问题：

- 在主机中只运行必要的服务。主机中运行的服务越少，安全性等级和性能也就越高。
- 在管理程序中启用 SELinux。有关使用 SELinux 和虚拟化的资料请阅读 [7.1 “SELinux 和虚拟化”](#)。
- 使用防火墙来限制 dom0 的流量。您可以设置采用 default-reject 规则的防火墙，这将有助于避免对 dom0 的攻击。限制网络 facing 服务也是很重要的。
- 不要允许普通用户访问 dom0。如果您允许普通用户访问 dom0，这可能会导致 dom0 出现薄弱环节。请记住，dom0 是专用的，允许非授权帐户的访问可能会降低安全等级。

### 7.1. SELinux 和虚拟化

安全性加强 Linux 是由 Linux 社区协助 NSA 开发，可为 Linux 提供更强大的安全性。SELinux 可限制攻击者的能力，并可阻止很多常见安全性漏洞，比如缓存溢出攻击以及特权升级。因为这些有点，Fedora 建议所有 Linux 系统应启用 SELinux 并在 enforcing 模式下运行。

如果启用了 SELinux 或者该映像不在正确目录中时，SELinux 会阻止载入客户端映像。SELinux 要求将所有客户端映像保存在 /var/lib/libvirt/images 目录中。

#### 在使用 enforcing 模式的 SELinux 中添加基于 LVM 的存储

下面的内容是在启用了 SELinux 的虚拟客户端中添加逻辑卷的示例。这些说明也可用于硬盘分区。

#### 过程 7.1.0 在启用了 SELinux 的虚拟客户端中创建并挂载逻辑卷

1. 创建逻辑卷。这个示例创建了一个 5GB 的逻辑卷，名为 `NewVolumeName`，位置在名为 `volumegroup` 的卷组中。

```
# lvcreate -n NewVolumeName -L 5G volumegroup
```

2. 使用支持扩展属性的文件系统格式化 `NewVolumeName` 逻辑卷，比如 ext3。

```
# mke2fs -j /dev/volumegroup/NewVolumeName
```

3. 为挂载新逻辑卷创建新目录。这个目录可在您文件系统的任意位置。建议您将其放在重要的系统目录（/etc、/var、/sys）或者主目录中（/home 或者 /root）。这个示例使用名为 /virtstorage 的目录。

```
# mkdir /virtstorage
```

4. 挂载逻辑卷。

```
# mount /dev/volumegroup/NewVolumeName /virtstorage
```

5. 为 Xen 文件夹设定正确的 SELinux 类型。

```
semanage fcontext -a -t xen_image_t "/virtualization(/.*)?"
```

另外为 KVM 文件夹设定正确的 SELinux 类型。

```
semanage fcontext -a -t virt_image_t "/virtualization(/.*)?"
```

如果使用对象策略（即默认策略），则该命令会在 `/etc/selinux/targeted/contexts/files/file_contexts.local` 文件中添加一行，以便使更改持久。添加的行可能类似如下：

```
/virtstorage(/.*)? system_u:object_r:xen_image_t:s0
```

6. 运行该命令将挂载点类型（`/virtstorage`）以及其中的所有文件改为 `xen_image_t`（`restorecon` 和 `setfiles` 读取 `/etc/selinux/targeted/contexts/files/` 中的文件）。

```
# restorecon -R -v /virtualization
```

## 7.2. SELinux 的相关事宜

这部分内容包含了在您的虚拟化环境中实施 SELinux 时需要考虑的问题。当您部署系统更改或者添加设备时，您必须相应地更新 SELinux 策略。要为客户端配置 LVM 卷，您必须为不同的基本块设备和卷组修改 SELinux 上下文。

```
# semanage fcontext -a -t xen_image_t -f -b /dev/sda2
# restorecon /dev/sda2
```

在重启守护进程后，可使用布尔值参数 `xend_disable_t` 在未修改的模式中设置 `xend`。最好是为单一守护进程而不是整个系统禁用此保护。建议您不要将广泛使用的目录重新标记为 `xen_image_t`。

## 网络配置

本章提供基于 libvirt 应用程序使用的通用联网配置说明。这个信息对所有管理程序，无论是 Xen、KVM 还是其它都适用。有关详情请参考 libvirt 网络构架文档。

两个通用设置为“虚拟网络”或者“共享物理设备”。前一个在所有发行本中都一致，可即时使用。后一个需要针对具体发行本进行手动配置。

### 8.1. libvirt 的网络地址转换（NAT）

共享网络连接最常用的方法之一就是使用网络地址转换（NAT）转发（也就是我们说的虚拟网络）。

#### 主机配置

每个标准 libvirt 安装都提供即时到虚拟机的基于 NAT 的连接性，这也被称为“默认虚拟网络”。使用 `virsh net-list --all` 命令验证其可用性。

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

如果缺少它，可重新载入并激活示例 XML 配置文件：

```
# virsh net-define /usr/share/libvirt/networks/default.xml
```

默认网络是在 `/usr/share/libvirt/networks/default.xml` 中定义的

将默认网络设定为自动启动：

```
# virsh net-autostart default
Network default marked as autostarted
```

启动默认网络：

```
# virsh net-start default
Network default started
```

libvirt 默认网络运行后，您将会看到独立的桥接设备。这个设备不能添加任何物理接口，因为它使用 NAT 和 IP 转发与外界连接，因此不要添加新接口。

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

libvirt 可添加 iptables 规则，该规则允许所有进入和来自附加到 virbr0 设备的客户端流量，该设备可处于 INPUT、FORWARD、OUTPUT 和 POSTROUTING 链中。libvirt 随后试图启用 `ip_forward` 参数。其它一些程序可能会禁用 `ip_forward`，因此最佳选择是在 `/etc/sysctl.conf` 中添加以下内容。

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

## 客户端配置

完成主机配置后，就可将客户端根据其名称连接到虚拟网络中。要将某个客户端连接到“默认”虚拟网络中，请在该客户端中使用以下 XML：

```
<interface type='network'>
  <source network='default' />
</interface>
```

### Note

定义 MAC 地址是可选的。如果跳过则自动生成 MAC 地址。手动设置 MAC 地址在某些情况下非常有用。

```
<interface type='network'>
  <source network='default' />
  <mac address='00:16:3e:1a:b3:4a' />
</interface>
```

## 8.2. 使用 libvirt 桥接的联网

桥接联网（也称为物理设备共享）是将物理设备专门用于一台虚拟机。桥接通常用于高级设置以及使用多个网络接口的服务器。

### 禁用 Xen 网络脚本

如果您的系统使用 Xen 桥接，则建议您编辑 `/etc/xen/xend-config.sxp` 文件并更改以下行禁用默认 Xen 网络桥接：

```
(network-script network-bridge)
```

至：

```
(network-script /bin/true)
```

### 禁用网络管理器 (NetworkManager)

网络管理器不支持桥接。必须禁用网络管理器才可使用旧的网络脚本联网。

```
# chkconfig NetworkManager off
# chkconfig network on
# service NetworkManager stop
# service network start
```

### Note

与其关闭网络管理器，您也可以在示例使用的 `ifcfg-*` 脚本中添加 `"NM_CONTROLLED=no"`。

## 生成网络初始化脚本

创建或者编辑以下两个网络配置文件。可重复这一步（使用不同名称）添加额外网络桥接。

改为 `/etc/sysconfig/network-scripts` 目录：

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

为您添加到桥接中的设备打开网络脚本。在这个示例中，`ifcfg-eth0` 定义已被设定为桥接一部分的物理网络接口：

```
DEVICE=eth0
# change the hardware address to match the hardware address your NIC uses
HWADDR=00:16:76:D6:C9:45
ONBOOT=yes
BRIDGE=br0
```



### 窍门

您可以在配置文件最后附加 `MTU` 变量配置设备的最大传输单位（MTU）。

```
MTU=9000
```

在 `/etc/sysconfig/network-scripts` 目录中创建名为 `ifcfg-br0` 和类似的新网络脚本。`br0` 是桥接名称，它可以是任何名称，只要该文件名称与 `DEVICE` 参数一致即可。

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=dhcp
ONBOOT=yes
DELAY=0
```



### Warning

The line, `TYPE=Bridge`, is case-sensitive. It must have uppercase 'B' and lower case 'ridge'.

配置后请重启联网或者重新引导。

```
# service network restart
```

Configure iptables to allow all traffic to be forwarded across the bridge.

```
# iptables -I FORWARD -m physdev --physdev-is-bridged -j ACCEPT
# service iptables save
# service iptables restart
```



### Disable iptables on bridges

Alternatively, prevent bridged traffic from being processed by iptables rules. In `/etc/sysctl.conf` append the following lines:

```
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
```

Reload the kernel parameters configured with `sysctl`

```
# sysctl -p /etc/sysctl.conf
```

Restart the libvirt daemon.

```
# service libvirtd reload
```

您现在应该有一个“共享物理设备”，它可访问客户端并对 LAN 有完全访问权限。验证您的新桥接：

```
# brctl show
bridge name    bridge id        STP enabled    interfaces
virbr0         8000,000000000000  yes
br0            8000,000e0cb30550  no            eth0
```

请注意：该桥接完全独立于 `virbr0` 桥接。▯▯试图将物理设备附加到 `virbr0`。 `virbr0` 桥接只用于网络地址转换（NAT）连接性。

## KVM 半虚拟驱动程序

半虚拟驱动程序可用于 KVM 主机中运行的虚拟 Windows 客户端。这些半虚拟驱动程序包含在 virtio 软件包中。Virtio 软件包支持块（存储）设备和网络接口控制器。

半虚拟驱动程序改进了全虚拟客户端的性能。使用半虚拟驱动程序可降低客户端 I/O 等待时间，流量可达到近似裸机的水平。建议您在运行很多 I/O 任务和程序的全虚拟客户端中使用半虚拟驱动程序。

新的 Fedora 版本中可自动载入并安装 KVM 半虚拟驱动程序。那些 Fedora 版本探测并安装这些驱动程序因此不需要额外的安装步骤。

与 KVM 模块一样，virtio 驱动程序只可用于运行较新版本 Fedora 的主机。



### Note

每个客户端只有 28 个 PCI 插槽可用于附加设备。每个半虚拟网络或者块设备使用一个插槽。每个客户端最多可使用 28 个附加设备，它们可以是任意组合的半虚拟网络、半虚拟磁盘设备或者其它使用 VTd 的 PCI 设备。

以下微软 Windows 版本支持 KVM 半虚拟驱动程序：

- Windows XP，
- Windows Server 2003，
- Windows Vista 以及
- Windows Server 2008。

### 9.1. 安装 KVM Windows 半虚拟驱动程序

这部分论述了如何安装 KVM Windows 半虚拟驱动程序。KVM 半虚拟驱动程序可在 Windows 安装过程中载入，也可在客户端安装后载入。

您可使用以下方法之一在您的主机中安装半虚拟驱动程序：

- 在客户端可访问的网络中托管安装文件，
- 使用驱动程序安装磁盘 .iso 的虚拟光驱设备或者
- 使用虚拟软盘设备在引导时安装这些驱动程序（用于 Windows 客户端）。

这个指南论述了将半虚拟安装程序磁盘作为虚拟光驱设备的安装。

#### 1. 下载驱动程序

微软提供这些驱动程序（[windowsservercatalog.com](http://windowsservercatalog.com)<sup>1</sup>）。

*virtio-win* 软件包在 `/usr/share/virtio-win/` 目录中安装安装光驱映像 `virtio-win.iso`。

#### 2. 安装半虚拟驱动程序

建议您在附加或者修改设备使其使用半虚拟驱动程序前在客户端中安装该驱动程序。

在保存根文件系统的块设备或者需要引导客户端的其它块设备中，必须在修改该设备前安装驱动程序。如果没有在客户端中安装驱动程序，但将驱动程序设定为 virtio 驱动程序，则无法引导该客户端。

### 使用 virt-manager 挂载映像

使用 virt-manager 添加光驱映像请参考 9.1, “virt-manager Windows ”。

#### 过程9.1. 使用virt-manager 在 Windows 客户端中挂载光驱映像

1. 打开 virt-manager，在虚拟机列表中选择您的虚拟客户端并按 详情 按钮。
2. 点击「详情」面板中的 添加 按钮。
3. 这样就打开了一个添加新设备向导。从下拉菜单中选择「存储设备」，然后点 前进。



4. 选择「文件（磁盘映像）」选项，并将半虚拟驱动程序文件设定为 .iso 文件。如果您使用 yum 安装半虚拟驱动程序软件包，那么这个 .iso 文件的位置就是 /usr/share/xenpv-win。

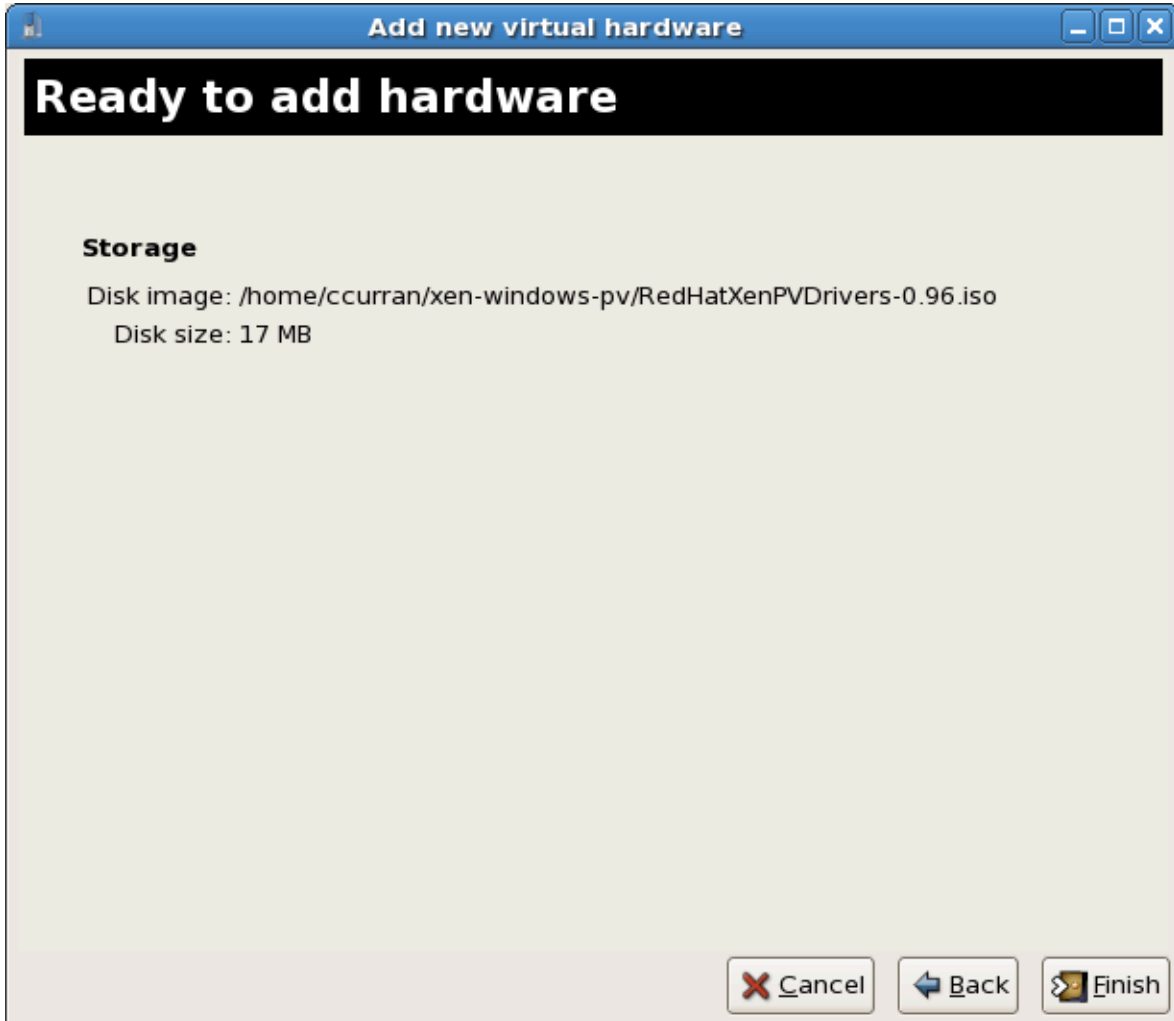
如果该驱动程序是保存在物理光盘中，请使用「普通磁盘分区」选项。

将「设备类型」设定为「IDE cdrom」并点击 前进 继续。





5. 该磁盘已分配，启动客户端后即可使用。请点击 完成 关闭向导，如果有错误操作请返回。



### 使用虚拟软盘安装

这个过程包括在 Windows 安装过程中安装半虚拟驱动程序。

- 第一次安装 Windows VM 时请使用立即运行菜单将 viostor.vfd 附加为一个软盘
  - a. Windows Server 2003  
当窗口提示按 F6 使用第三方驱动程序时，请按 F6 并根据屏幕指示操作。
  - b. Windows Server 2008  
当安装程序提示您安装驱动程序时，请点击“载入驱动程序”将安装程序指向驱动器 A: 并选择适合您的操作系统和脚本（bittage）的驱动程序

### 在现有设备中使用 KVM 半虚拟驱动程序

修改现有附加到客户端中的硬盘使其使用 virtio 驱动程序而不是虚拟 IDE 驱动程序。这个示例编辑了 libvirt 配置文件。另外使用 virt-manager、virsh attach-disk 或者 virsh attach-interface 可添加使用半虚拟驱动程序的新设备 `virtio KVM`。

1. 以下是使用虚拟 IDE 驱动程序的基于文件的块设备。这是用于不使用虚拟驱动程序的虚拟客户端的典型条目。

```
<disk type='file' device='disk'>  
  <source file='/var/lib/libvirt/images/disk1.img'/>  
  <target dev='hda' bus='ide'/>  
</disk>
```

2. 将 bus= 条目改为 virtio 可让该条目使用半虚拟设备。

```
<disk type='file' device='disk'>  
  <source file='/var/lib/libvirt/images/disk1.img'/>  
  <target dev='hda' bus='virtio'/>  
</disk>
```

### 在新设备中使用 KVM 半虚拟驱动程序

这个步骤是用 virt-manager 创建使用 KVM 半虚拟驱动程序的新设备。

另外，使用 virsh attach-disk 或者 virsh attach-interface 命令可附加使用半虚拟驱动程序的设备。



#### 首先安装驱动程序

请在 Windows 客户端中安装新设备前确定安装了该驱动程序。如果该驱动程序不可用，则无法识别和使用该设备。

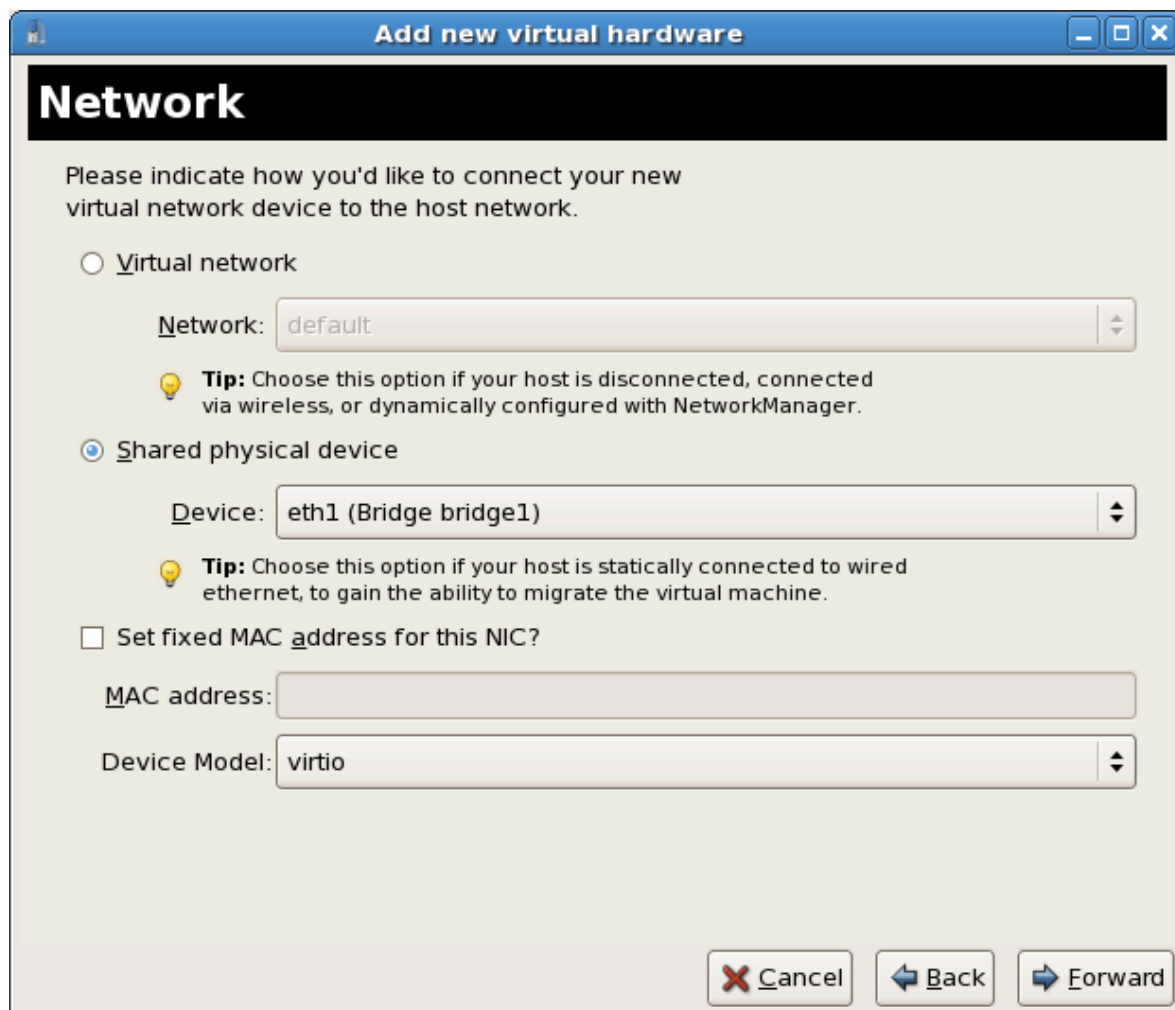
1. 请在 virt-manager 中双击客户端名称打开虚拟客户端。
2. 打开「硬件」标签。
3. 按「添加硬件」按钮。
4. 在“添加虚拟硬盘”标签中选择「存储」或者「网络」作为设备类型。
  1. 新磁盘设备

选择存储设备或者基于文件的映像。选择「Virtio 磁盘」作为「设备类型」并按「前进」。

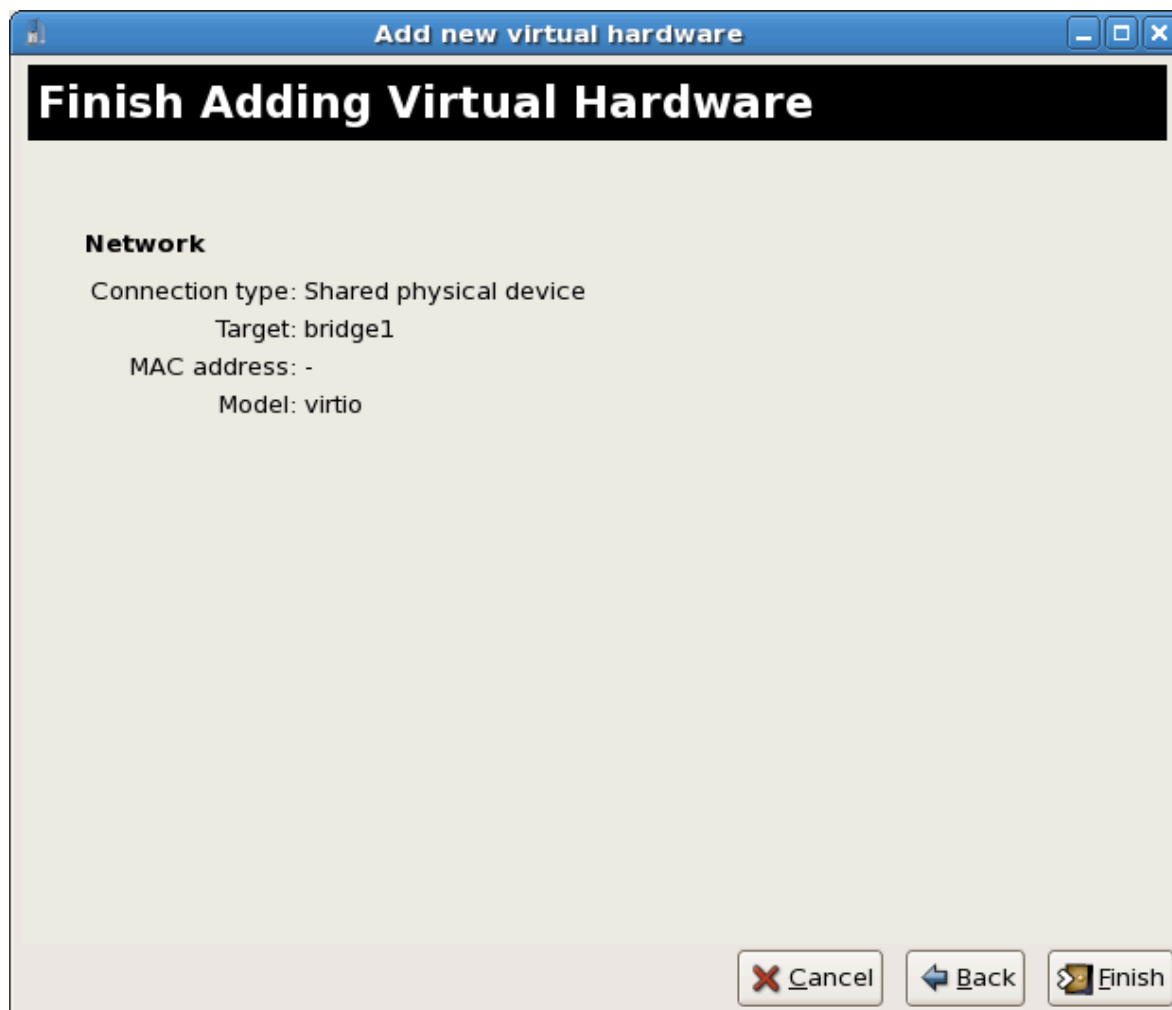


## 2. 新网络设备

选择「虚拟网络」或者「共享物理设备」。选择「virtio」作为「设备类型」并按「前进」。



5. 按「完成」保存该设备。



6. 重启该客户端。Windows 客户端要在重启后才可识别该设备。

---

## 部分 III. Administration

### 管理虚拟系统

这些章节包含使用 Fedpra 中的工具管理主机及虚拟客户端的信息。

---

---



## 用 xend 管理客户端

xend 节点控制守护进程执行某些与虚拟机相关的系统管理功能。这个守护进程控制虚拟资源，而且必须运行 xend 与虚拟机互动。在您启动 xend 之前，您必须编辑 xend 的配置文件 `xend-config.sxp` 来指定操作参数。下面是您可以在 `xend-config.sxp` 配置文件里启用或禁用的参数：

项目	Description
(console-limit)	决定控制台服务器的内存缓冲限制 <code>xend_unix_server</code> ，并根据域分配数值。
(min-mem)	决定为 <code>domain0</code> 保留的最小内存数量（以 MB 为单位），如果为 0，则值不变化。
(dom0-cpus)	决定 <code>domain0</code> 使用的 CPU 数量（默认情况下至少分配一个 CPU）。
(enable-dump)	决定当发生崩溃时启用转储（默认为 0）。
(external-migration-tool)	决定用来处理外部设备迁移的脚本或应用程序。脚本必须位于 <code>etc/xen/scripts/external-device-migrate</code> 。
(logfile)	决定日志文件的位置（默认为 <code>/var/log/xend.log</code> ）。
(loglevel)	过滤日志模式值：DEBUG、INFO、WARNING、ERROR 或 CRITICAL（默认为 DEBUG）。
(network-script)	决定启用联网环境的脚本（脚本必须位于 <code>etc/xen/scripts</code> 目录）。
(xend-http-server)	启用 <code>http stream</code> 数据包管理服务器（默认为 no）。
(xend-unix-server)	启用 <code>unix</code> 域套接字服务器（套接字服务器是一个通信终点，它处理底层的网络连接以及接受或拒绝转入的连接）。默认值为 yes。
(xend-relocation-server)	启用用于跨机器迁移的重定向服务器（默认为 no）。
(xend-unix-path)	指定 <code>xend-unix-server</code> 命令输出数据的位置（缺省是 <code>var/lib/xend/xend-socket</code> ）
(xend-port)	决定 <code>http</code> 管理服务器使用的端口（默认为 8000）。
(xend-relocation-port)	决定重定向服务器所使用的端口（默认为 8002）。
(xend-relocation-address)	决定允许迁移的主机地址。默认值为 <code>xend-address</code> 。
(xend-address)	决定域套接字服务器绑定的地址。默认为允许所有连接。

表 10.1. xend 配置参数

在设置了这些操作参数后，您应该确定 xend 是否正在运行，如果没有，就初始化该守护进程。在命令提示符后，您可以用下面的命令启动 xend 守护进程：

```
service xend start
```

你可以用 xend 来停止这个守护进程：

```
service xend stop
```

这个命令停止了守护进程。

你也可以用 xend 来重新启动守护进程：

```
service xend restart
```

守护进程再次启动了。

你可以检查 xend 守护进程的状态。

```
service xend status
```

下面的输出显示了守护进程的状态。



### 在引导时启用 xend

使用 `chkconfig` 命令在 `init.script` 中添加 xend。

```
chkconfig --level 345 xend
```

现在会在运行级别 3、4 和 5 中启动 xend。

## KVM 客户端计时管理

KVM 使用很多先进 CPU 的恒定时间戳计数器（TSC）特性。一些没有恒定时间戳计数器的 CPU 会影响 KVM 中运行的客户端计时。无法准确计时的客户端可能会严重影响一些联网应用程序，因为您的客户端可能会比实际时间运行得快或者慢。

不准确的时钟和计数器为客户端带来的一些问题：

- 时钟可能会无法与实际时间同步，这将导致会话无效并影响到网络。
- 时钟慢一些的客户端迁移时会出现问题。
- 客户端可能会停止或者崩溃。

这些问题在其它虚拟化平台中也存在，因此应经常测试计时。



### NTP

应该在主机和客户端中运行网络时间协议（NTP）守护进程。请启用 `ntpd` 服务：

```
# service ntpd start
```

请在默认启动序列中添加 `ntpd` 服务：

```
# chkconfig ntpd on
```

使用 `ntpd` 服务应该可以在各种情况下最小化时钟偏移造成的影响。

### 确定您的 CPU 是否有恒定时间戳计数器

如果显示 `constant_tsc` 标签，那么您的 CPU 就有恒定时间戳计数器。要确定您的 CPU 是否有 `constant_tsc` 标签，请运行以下命令：

```
$ cat /proc/cpuinfo | grep constant_tsc
```

如果有输出结果，说明您的 CPU 有 `constant_tsc` 字节。如果没有输出结果，请按以下步骤执行。

### 配置没有恒定时间戳计数器的主机

没有恒定时间戳计数器的系统需要进行额外配置。电源管理特性可影响准确计时，且必须在客户端中禁用以便使用 KVM 准确计时。



### Note

这些说明仅用于 AMD 修订 F cpu。

如果 CPU 缺少 `constant_tsc` 字节，则请禁用所有电源管理特性（[BZ#513138](https://bugzilla.redhat.com/show_bug.cgi?id=513138)<sup>1</sup>）。每个系统都有多个用来计时的计时器。TSC 在主机中不稳定，有时是由于更改 `cpufreq`、`deep C state` 或者迁移到拥有更

<sup>1</sup> [https://bugzilla.redhat.com/show\\_bug.cgi?id=513138](https://bugzilla.redhat.com/show_bug.cgi?id=513138)

迅速 TSC 的主机所致。要停止 deep C state，即停止 TSC，请在主机 grub 的内核引导选项中添加 "processor\_max\_cstate=1"：

```
term Fedora (vmlinuz-2.6.29.6-217.2.3.fc11)
  root (hd0,0)
  kernel /vmlinuz-vmlinuz-2.6.29.6-217.2.3.fc11 ro root=/dev/VolGroup00/LogVol00 rhgb
  quiet processor_max_cstate=1
```

要禁用 cpufreq（只有在没有 cpufreq 的主机中需要），请编辑 /etc/sysconfig/cpuspeed 配置文件并将 MIN\_SPEED 和 MAX\_SPEED 变量改为可用的最高频率。有效上限请查看 /sys/devices/system/cpu/cpu\*/cpufreq/scaling\_available\_frequencies 文件。

### 在红帽企业版 Linux 客户端中使用半虚拟时钟

某些红帽企业版 Linux 客户端需要附加内核参数。这些参数的设置可通过在客户端 /boot/grub/grub.conf 文件的 /kernel 行结尾添加它们完成。

下表列出了红帽企业版 Linux 版本以及没有恒定时间戳计数器客户端所需要的参数。

红帽企业版 Linux	附加客户端内核参数
使用半虚拟时钟的 5.4 AMD64/ Intel 64	不需要附加参数
没有半虚拟时钟的 5.4 AMD64/ Intel 64	divider=10 notsc lpj=n
使用半虚拟时钟的 5.4 x86	不需要附加参数
没有半虚拟时钟的 5.4 x86	divider=10 clocksource=acpi_pm lpj=n
5.3 AMD64/Intel 64	divider=10 notsc
5.3 x86	divider=10 clocksource=acpi_pm
4.8 AMD64/Intel 64	notsc divider=10
4.8 x86	clock=pmtmr divider=10
3.9 AMD64/Intel 64	不需要附加参数
3.9 x86	不需要附加参数

### 在 Windows 客户端中使用半虚拟时钟

在 Windows 客户端中编辑引导参数可启用半虚拟时钟。Windows 引导设置保存在 boot.ini 文件中。要弃用半虚拟时钟，请添加以下行：

```
/use pmtimer
```

有关 Windows 引导设置以及 pmtimer 选项的详情请参考 [Windows XP 的 boot.ini 文件](#) 或 [Windows 2003 boot.ini 文件](#)<sup>2</sup>。

<sup>2</sup> <http://support.microsoft.com/kb/833721>

## KVM 实时迁移

本章论述了如何将 KVM 管理程序中运行的客户端迁移到其它 KVM 主机中。

迁移就是将虚拟客户端从一个主机移动到另一个主机的进程名称。迁移是虚拟化的一个主要特性，因为软件是与硬件完全分开的。迁移在以下方面很有用：

- Load balancing - guests can be moved to hosts with lower usage when a host becomes overloaded.
- Hardware failover - when hardware devices on the host start to fail, guests can be safely relocated so the host can be powered down and repaired.
- Energy saving - guests can be redistributed to other hosts and host systems powered off to save energy and cut costs in low usage periods.
- Geographic migration - guests can be moved to another location for lower latency or in serious circumstances.

迁移可离线或者实时进行。要迁移客户端，则存储必须是共享的。迁移工作是将客户端内存发送到目的主机中。共享存储保存客户端的默认文件系统。该文件系统映像不是通过网络从源主机发送到目的主机的。

An offline migration suspends the guest then moves an image of the guests memory to the destination host. The guest is resumed on the destination host and the memory the guest used on the source host is freed.

离线迁移所需时间依赖网络带宽和等待时间。有 2GB 内存的客户端在 1Gbit 以太网链接中平均需要十秒左右。

实时迁移可让客户端继续在源主机中运行并在不停止客户端的情况下开始移动内存。发送映像的同时监控所有修改的内存页面的变化并将其发送到目的地。使用更改的页面更新内存。持续进行此过程直到客户端允许的暂停时间与预计最后几页的传输时间相等。KVM 将估计剩余时间并尝试从源主机到目的主机传输大量的页面文件，直到 KVM 预计剩余的页面在虚拟客户端暂停的时候可以在很短时间内完成传输。在新主机中载入这些记录器，然后在目的地主机中恢复该客户端。如果无法合并该客户端（客户端处于极限负载时出现），则会暂停该客户端，然后启动离线迁移。

离线迁移所需时间依赖网络带宽和等待时间。如果网络正处于高负载或者带宽较小，则迁移会需要很长时间。

### 12.1.0 动态迁移要求

迁移客户端需要以下条件：

#### 迁移要求

- 使用以下协议之一在共享联网存储中安装虚拟客户端：
  - Fibre Channel
  - iSCSI
  - NFS
  - GFS2

- 同一版本的两个或者两个以上的 Fedora 系统有同样的更新。
- 两个系统必须打开正确的端口。
- 两个系统必须有一致的网络配置。两个主机中的所有桥接和网络配置必须完全一致。
- 源系统和目的系统中的共享存储必须挂载在同一位置。挂载的目录名必须一致。

### 配置网络存储

配置共享存储并在共享存储中安装客户端。共享存储说明请参考 [5.0 网络存储](#)。

另外，还可使用 NFS，示例参考 [12.2 “使用 NFS 进行简单迁移”](#)。

## 12.2.0 共享存储示例：使用 NFS 进行简单迁移

这个示例使用 NFS 与其它 KVM 主机共享客户端映像。这个示例不能在大量安装中使用，只用来演示迁移技术以及小规模部署。不要使用这个示例进行迁移或者运行多个虚拟客户端。

高级和更多鲁棒共享存储说明请参考 [5.0 网络存储](#)

1. 导出您的 libvirt 映像目录  
在 `/etc/exports` 文件中添加默认映像目录：

```
/var/lib/libvirt/images *.bne.redhat.com(rw,no_root_squash,async)
```

根据您的环境需要更改主机参数。

2. 启动 NFS
  - a. 如果没有安装请安装 NFS 软件包：

```
# yum install nfs
```

- b. 在 iptables 中为 NFS 打开端口并在 `/etc/hosts.allow` 文件中添加 NFS。
- c. 启动 NFS 服务：

```
# service nfs start
```

3. 在目的系统挂载共享存储  
在目的系统中挂载 `/var/lib/libvirt/images` 目录：

```
# mount sourceURL:/var/lib/libvirt/images /var/lib/libvirt/images
```



### 在源系统和目的系统中的位置必须一致

无论选择什么目录，客户端和主机的目录必须完全一致。这一点适用于所有共享存储类型。该目录必须一致否则迁移将会失败。

## 12.3. 使用 virsh 进行实时 KVM 迁移

使用 virsh 命令可将客户端迁移到另一台主机中。migrate 命令接受以下格式的参数：

```
# virsh migrate --live GuestName DestinationURL
```

The *GuestName* parameter represents the name of the guest which you want to migrate.

The *DestinationURL* parameter is the URL or hostname of the destination system. The destination system must run the same version of Fedora, be using the same hypervisor and have libvirt running.

Once the command is entered you will be prompted for the root password of the destination system.

### 示例：使用 virsh 进行实时迁移

这个示例是从 test1.bne.redhat.com 迁移至 test2.bne.redhat.com。请为您的环境更改主机名。这个示例迁移了名为 CentOS4test 的虚拟机。

这个示例假设您使用完整配置的共享存储并达到所有条件（列表如下：□□□□）。

#### 1. 确定客户端正在运行

在源系统 test1.bne.redhat.com 中确定 CentOS4test 正在运行：

```
[root@test1 ~]# virsh list
Id Name          State
-----
 10 CentOS4      running
```

#### 2. 迁移客户端

执行以下命令实时将客户端迁移至目的地 test2.bne.redhat.com。在目的地 URL 结尾添加 /system 告知 libvirt 您需要完全访问权限。

```
# virsh migrate --live CentOS4test qemu+ssh://test2.bne.redhat.com/system
```

Once the command is entered you will be prompted for the root password of the destination system.

#### 3. 等待

根据负载和客户端的大小，迁移可能需要一些时间。virsh 只报告出错信息。客户端仍将在源主机中继续运行直至全部迁移完成。

#### 4. 确定客户端已到达目的主机

在目的系统 test2.bne.redhat.com 中确认 CentOS4test 正在运行：

```
[root@test2 ~]# virsh list
Id Name          State
-----
 10 CentOS4      running
```

实时迁移完成。

**其它联网方法**  
libvirt 支持各种联网方法，其中包括 TLS/SSL、unix 套接字、SSH 以及不加密的 TCP。有关使用其它方法的详情请参考 13。

## 12.4. 使用 virt-manager 迁移

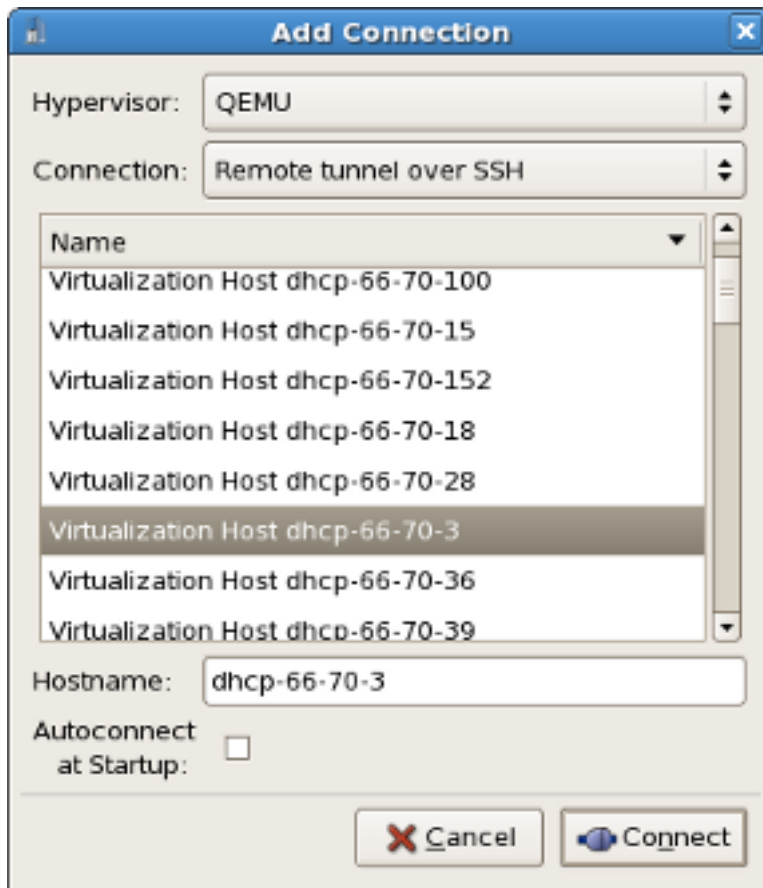
这部分论述了如何使用 virt-manager 迁移使用 KVM 的客户端。

1. 连接源和目标主机。在「文件」菜单中点击「添加连接」，此时会出现「添加连接」窗口。

输入以下详情：

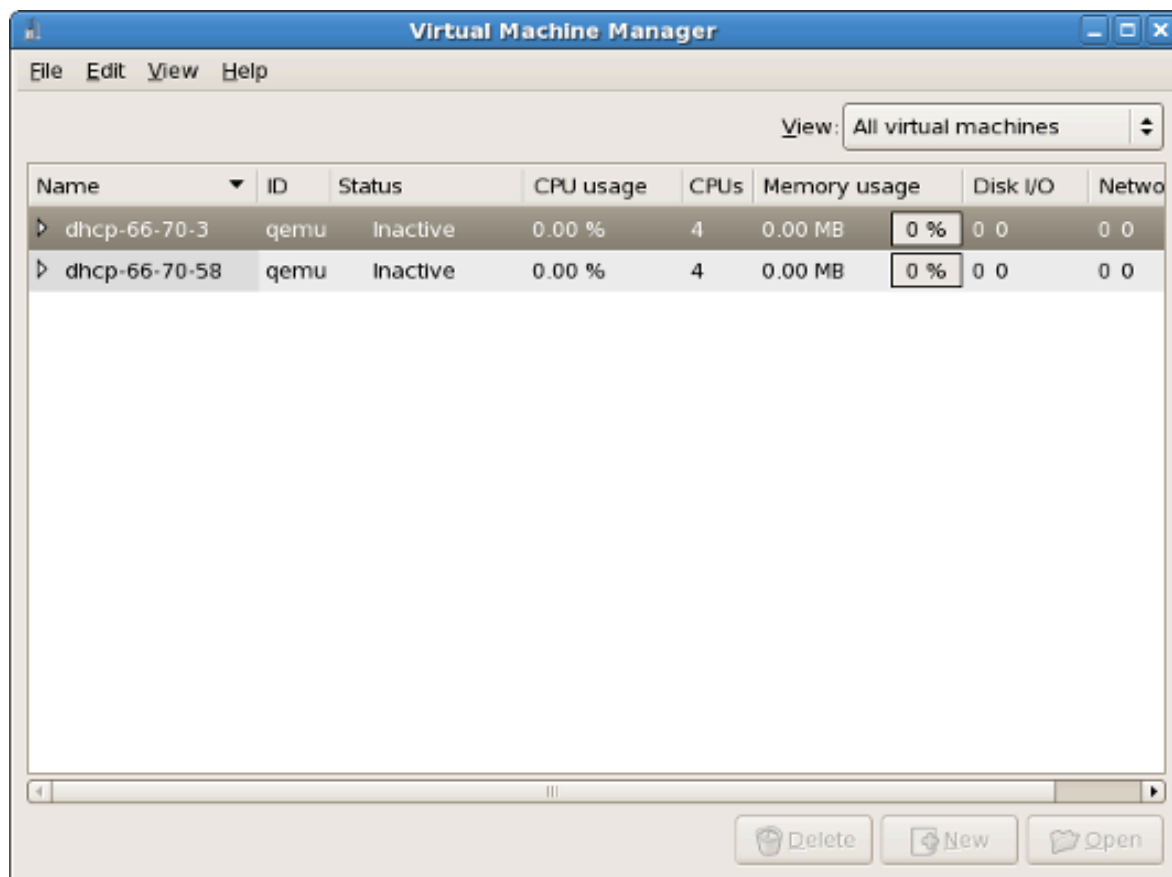
- 「管理程序」：请选择「QEMU」。
- 「连接」：选择连接类型。
- 「主机名」：输入主机名。

点击 连接。



虚拟机管理器显示连接的主机列表。

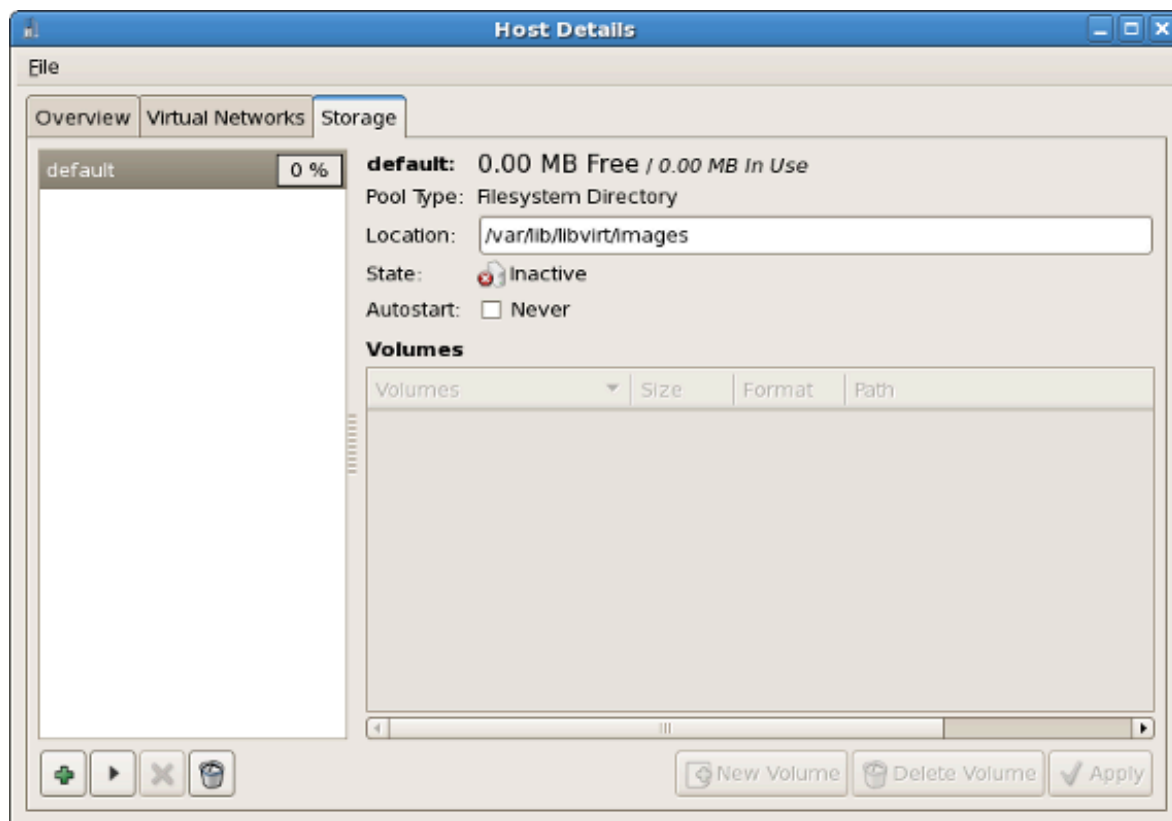




2. 在源主机和目标主机中添加有同样 NFS 的存储池。

在「编辑」菜单中点击「主机详情」，此时会出现主机详情窗口。

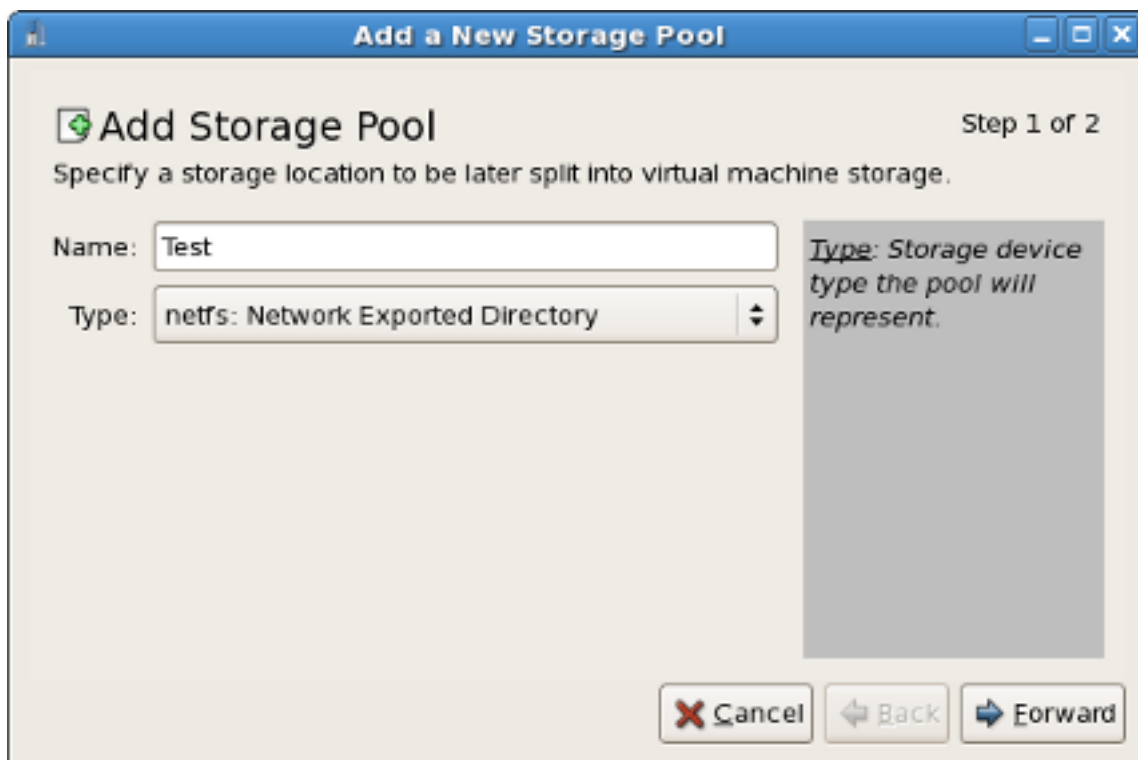
点击 存储 标签。



3. 添加一个新的存储池。在窗口的左下角点击 + 按钮。此时会出现添加新存储池窗口。

输入以下详情：

- 「名称」：输入存储池名称。
- 「类型」：选择「netfs：网络导出的目录」。



点击 前进。

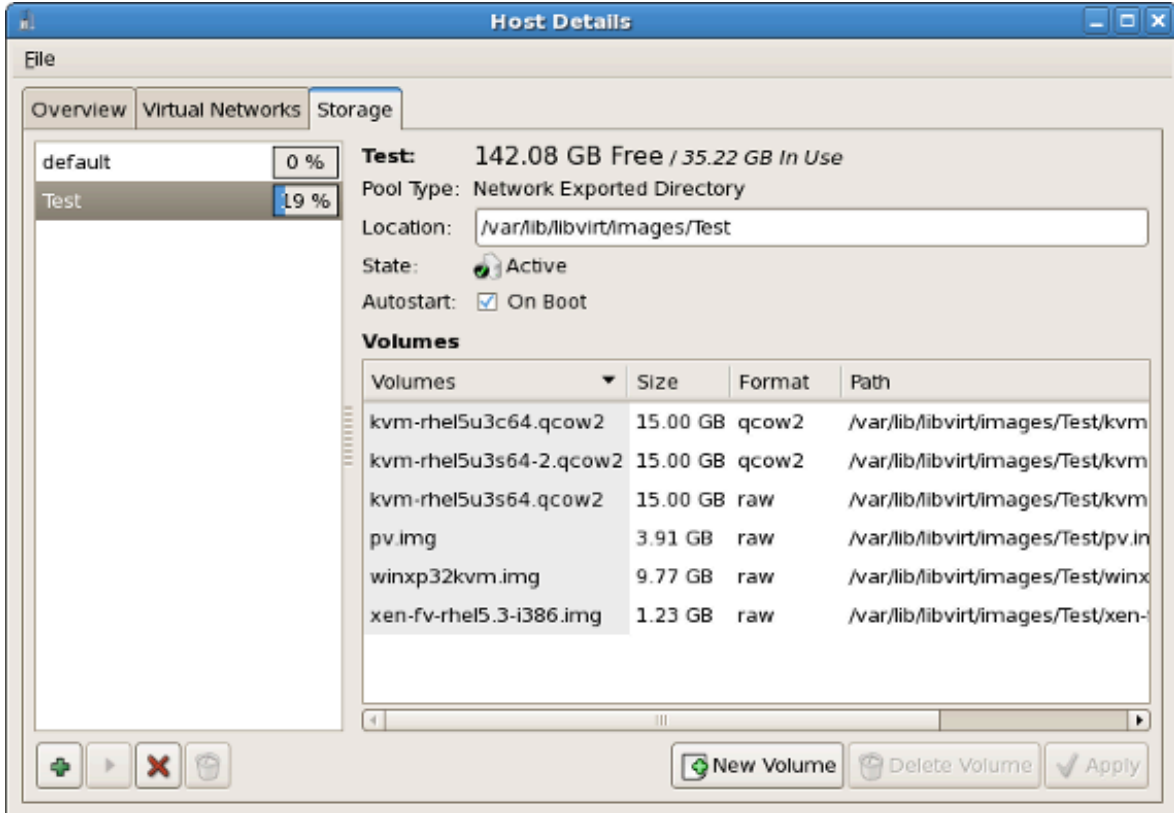
4. 输入以下详情：

- 「格式」：选择存储类型。实时迁移必须是 NFS 或者 iSCSI。
- 「主机名称」：输入存储服务器的 IP 地址或者完全限定域名。

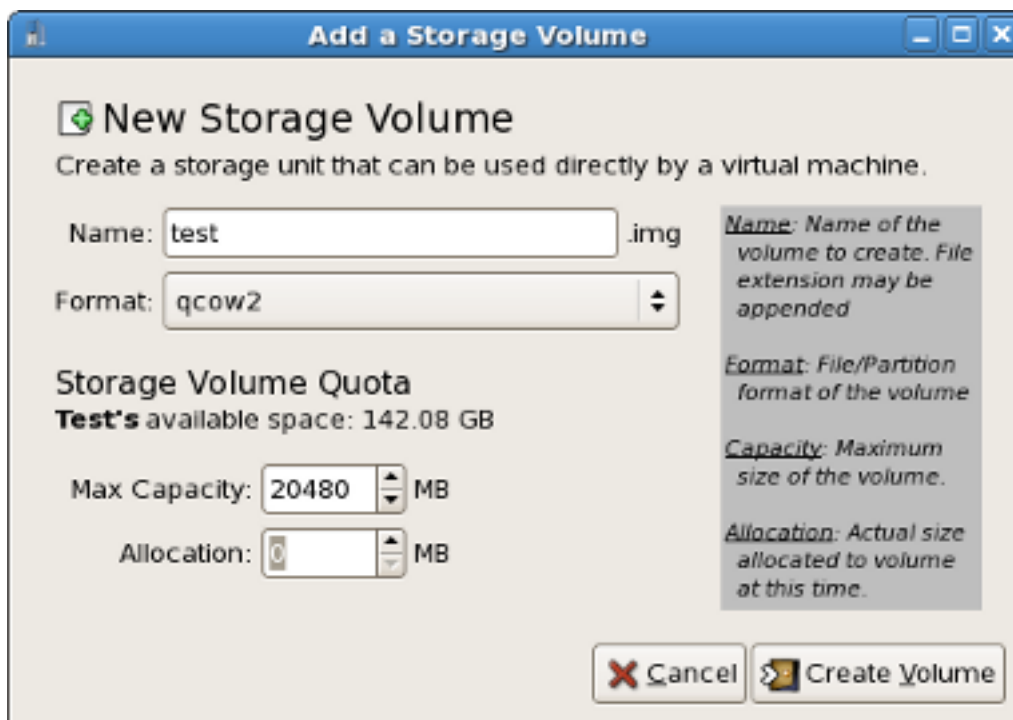


点击 完成。

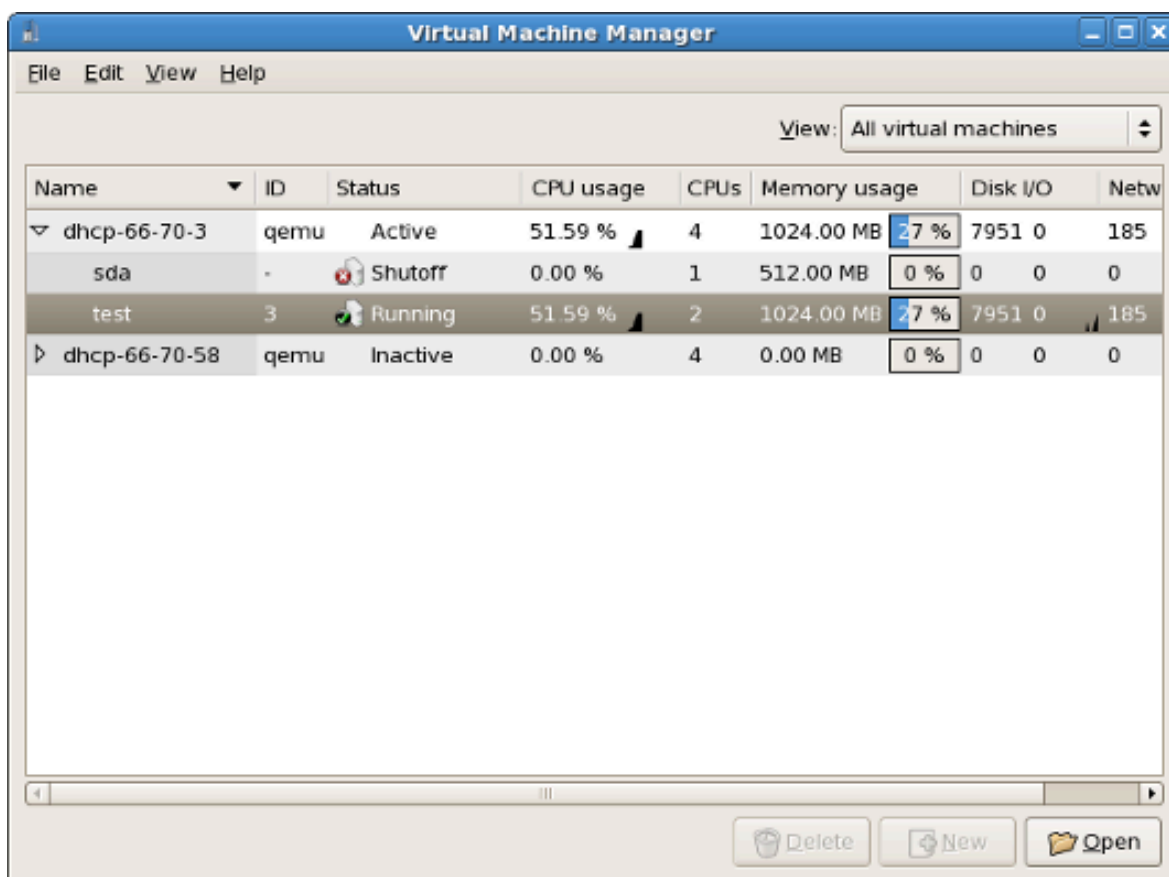
5. 在共享存储池中创建一个新卷，点击 新建卷。



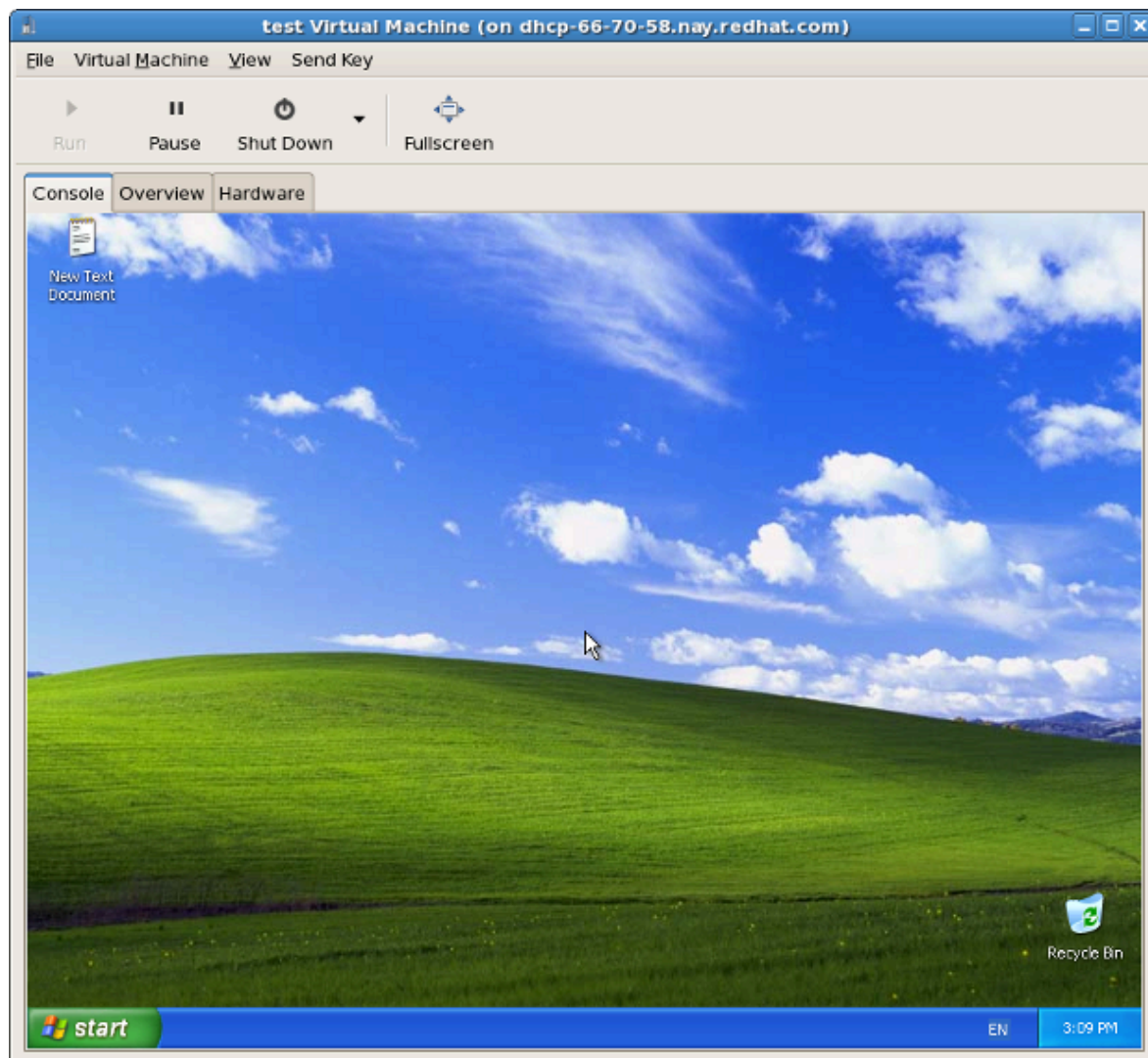
6. 输入详情然后点击 创建卷。



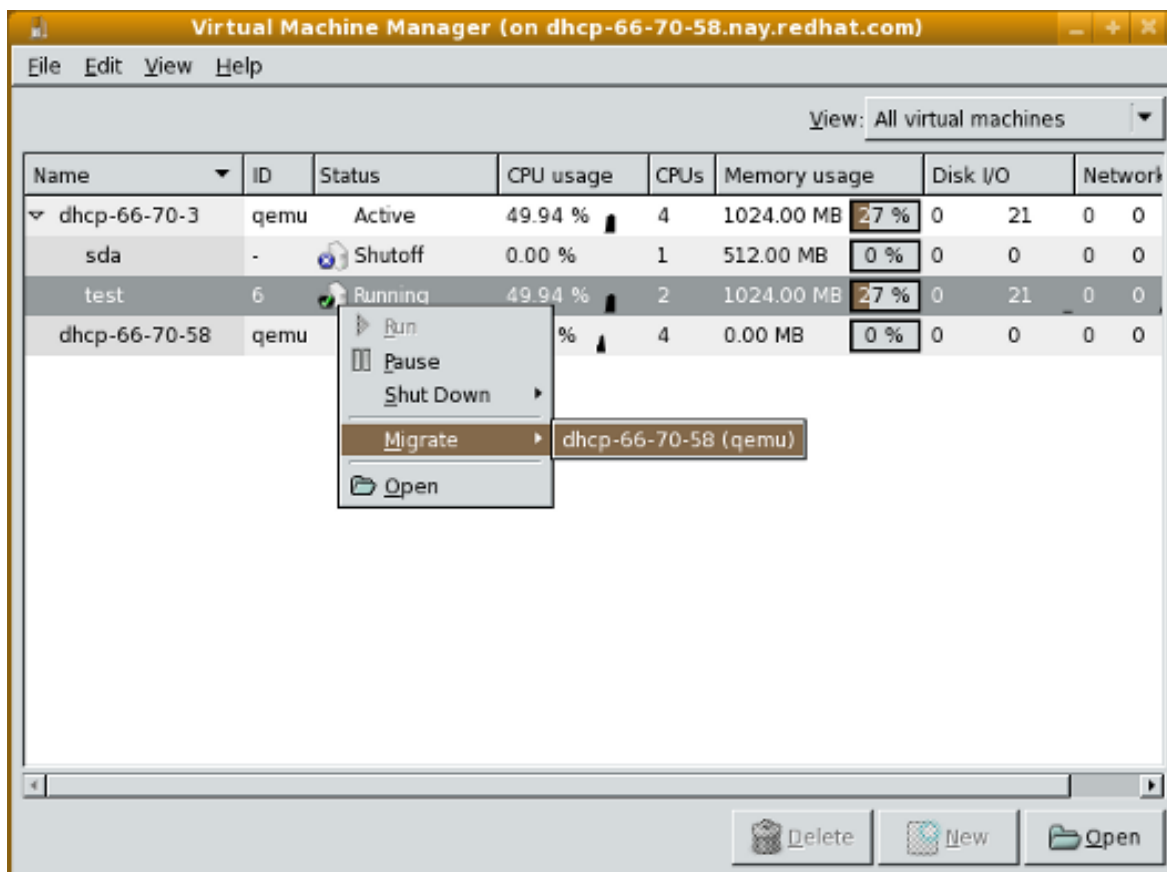
7. 使用新卷创建虚拟机，然后运行该虚拟机。



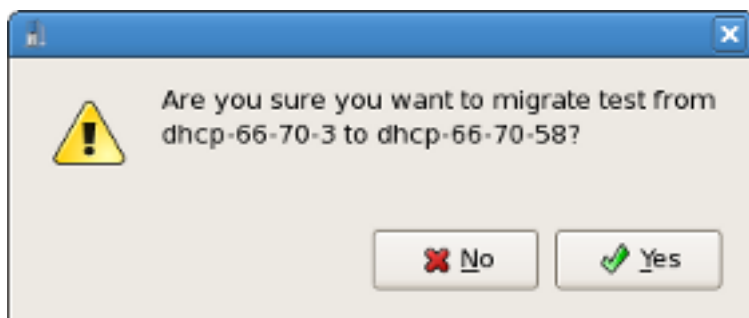
此时会出现虚拟机窗口。



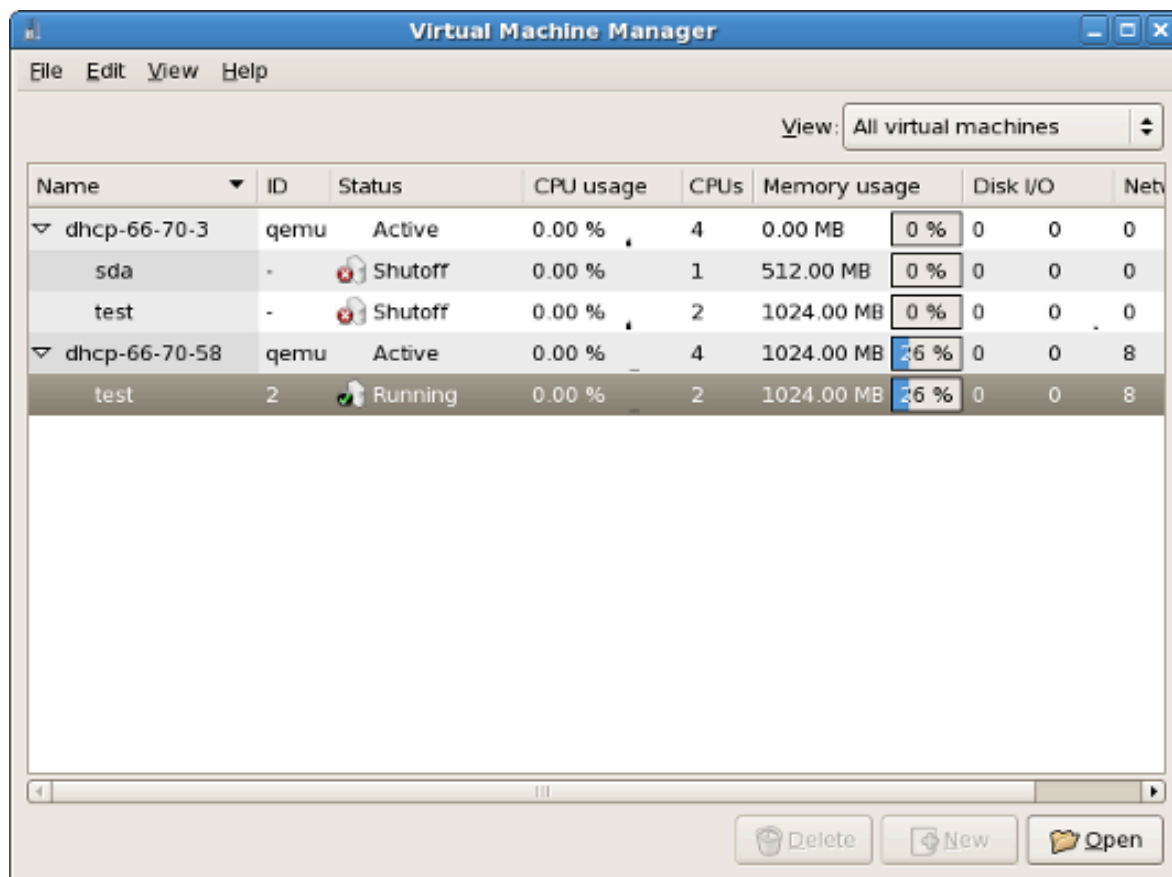
8. 在虚拟机管理器窗口中右键点击虚拟机，选择「迁移」，然后点击迁移位置。



9. 点击  是 确定迁移。

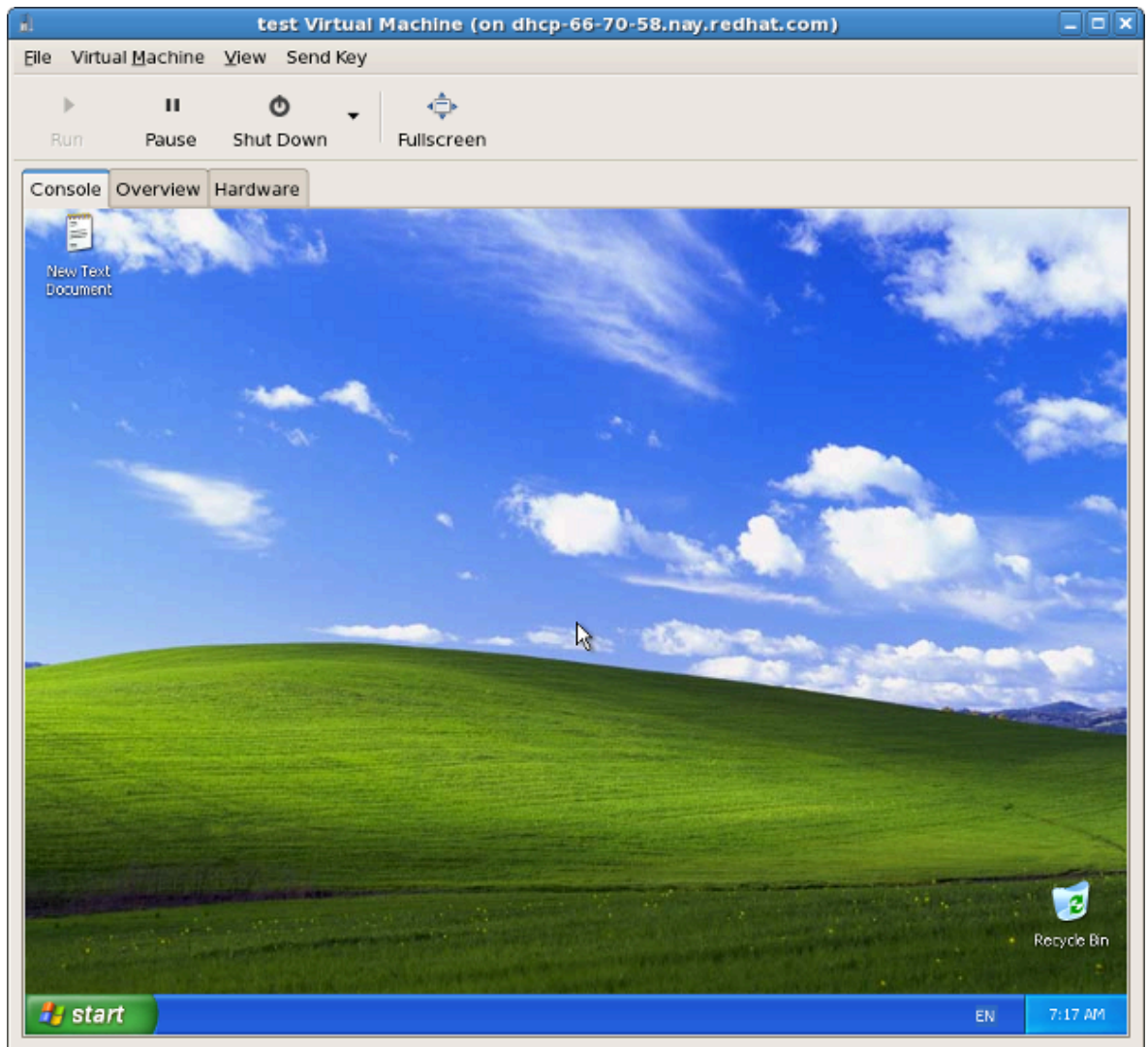


虚拟机管理器在其新位置显示虚拟机。



虚拟机管理器窗口显示新虚拟机的位置。







# 虚拟客户端的远程管理

这部分将论述如何使用 ssh、TLS 和 SSL 远程管理您的虚拟化客户端。

## 13.1. 使用 SSL 进行远程管理

ssh 软件包提供加密网络协议，可向远程虚拟化服务器安全发送管理功能。本方法使用 libvirt 通过 SSH 连接安全管理远程机器。所有验证都通过 SSH 公钥加密法和密码或者由您的本地 SSH 代理获得的密码短语进行。另外，每个客户端虚拟机的 VNC 控制台都将通过 SSH 连接。

SSH 通常是默认配置的，这样您就可以设定 SSH 密钥，并且不需要额外的防火墙规则访问管理服务或者 VNC 控制台。

在使用 SSH 对您的虚拟机进行远程管理时需要注意几点，它们是：

- 您需要以根用户身份登录访问远程机器来管理虚拟机，
- 最初的连接设定过程可能很慢，
- 没有一种标准或者简单的方法来撤销所有主机或者客户端中的用户密钥，且
- ssh 在处理大量远程机器时效果并不好。

为 `virt-manager` 配置 SSH 访问。

以下操作假设您是从头开始的，且还没有设定 SSH 密钥。

1. 您需要在您运行 `virt-manager` 的机器中有一个公钥对。如果您已经配置了 ssh，请跳过这个命令。

```
$ ssh-keygen -t rsa
```

2. 要允许远程登录，`virt-manager` 需要在每个运行 `libvirt` 的远程机器中有一个公钥副本。请使用 `scp` 命令从您要用来进行远程管理的机器中复制文件 `$HOME/.ssh/id_rsa.pub`：

```
$ scp $HOME/.ssh/id_rsa.pub root@somehost:/root/key-dan.pub
```

3. 文件复制完成后，请以根用户身份使用 `ssh` 命令连接到远程机器并将您复制的文件添加到认证的密钥列表中。如果远程主机中的根用户还没有认证的密钥列表，请确定正确设定了该文件的权限。

```
$ ssh root@somehost
# mkdir /root/.ssh
# chmod go-rwx /root/.ssh
# cat /root/key-dan.pub >> /root/.ssh/authorized_keys
# chmod go-rw /root/.ssh/authorized_keys
```

### libvirt 守护进程 (libvirtd)

libvirt 守护进程提供管理虚拟机的界面。您应该使用安装的 `libvirtd` 守护进程并在每个需要您管理的远程主机中运行该进程。使用 Fedora `kernel-xen` 软件包需要 `speci TODO`。

```
$ ssh root@somehost
# chkconfig libvirtd on
```

```
# service libvirtd start
```

配置完 libvirt 和 SSH 后，您应该可以远程访问并管理您的虚拟机。此时您还应该可以使用 VNC 访问您的客户端。

## 13.2. 使用 TLS 和 SSL 进行远程管理

您可以使用 TLS 和 SSL 管理虚拟机。TLS 和 SSL 提供更高的灵活性，但要比 ssh 复杂很多（请参考 [13.1 “SSL 认证”](#)）。TLS 和 SSL 是网页浏览器用来进行安全连接的同一个技术。libvirt 管理连接将为进入的连接打开一个 TCP 端口，这些连接是安全加密的，且根据 x509 证书进行验证。除 VNC 控制台外，还会将每个客户端虚拟机设定为使用带 x509 证书验证的 TLS。

这个方法不需要在要进行管理的远程机器中有 shell 帐户，但需要为访问管理服务或者 VNC 控制台设定额外的防火墙规则。证书取消列表可用来取消对用户的访问。

### 为 virt-manager 设定 TLS/SSL 访问的步骤

以下简短指南假定您从头开始设置，而且您没有任何 TLS/SSL 证书知识背景。如果您刚好有证书管理服务，那么您可以跳过第一步。

#### libvirt 服务器设定

有关生成证书的详情请参考 libvirt 网页 <http://libvirt.org/remote.html>。

#### Xen VNC 服务器

Xen VNC 服务器可以通过编辑配置文件 `/etc/xen/xend-config.sxp` 启用 TLS。删除配置文件 (`vnc-tls 1`) 配置参数前的注释标记。

`/etc/xen/vnc` 目录需要以下三个文件：

- `ca-cert.pem` - CA 证书
- `server-cert.pem` - CA 签注的服务器证书
- `server-key.pem` - 服务器私钥

这样就提供了数据频道的加密。可能会要求客户端出示其自身 x509 证书作为验证方法。要启用这个功能，请删除 (`vnc-x509-verify 1`) 参数的注释符号。

#### virt-manager 和 virsh 客户端设定

此时的客户端设定有一些不同。要通过 TLS 启用 libvirt 管理 API，就必须将 CA 以及客户端证书放在 `/etc/pki`。有关详情请参考 <http://libvirt.org/remote.html>。

在 virt-manager 用户界面，连接到主机时请使用 'SSL/TLS' 传送机制。

virsh 中 URL 有以下格式：

- 在 KVM 中使用 `qemu://hostname.guestname/system`。
- 在 Xen 中使用 `xen://hostname.guestname/`。

要为 VNC 启用 SSL 和 TLS，需要您将证书验证以及客户端证书放在 `$HOME/.pki` 中，即以下三个文件：

- CA 或者 `ca-cert.pem` - CA 证书。
- `libvirt-vnc` 或者 `clientcert.pem` - CA 签注的客户端证书。

- libvirt-vnc 或者 clientkey.pem - 客户端私钥。

## 13.3. 传输模式

对于远程管理，libvirt 支持以下传输模式：

### 传输层安全性 (TLS)

使用传输层安全性 TLS (SSL 3.1) 认证和加密的 TP/IP 插槽通常侦听在公用端口号上进行侦听。要使用这个方法，您需要生成客户端和服务端证书。标准端口为 16514。

### UNIX 插槽

Unix 域插槽只能从本地机器访问。插槽是不加密的，并使用 UNIX 权限或者 SELinux 进行认证。标准插槽名称为 /var/run/libvirt/libvirt-sock 和 /var/run/libvirt/libvirt-sock-ro (用于只读连接)。

### SSH

通过安全 Shell 协议 (SSH) 连接传输。需要安装 Netcat (*nc* 软件包)。必须在远程机器中运行 libvirt 守护进程 (libvirtd)。必须为 SSH 访问打开端口 22。您应该使用某些 ssh 密钥管理 (例如：ssh-agent 程序) 或者提示您输入密码。

### ext

ext 参数是可用于 libvirt 之外生成连接到远程机器的任意外部程序。这通常包括第三程序以及不支持的安全程序。

### tcp

未加密的 TCP/IP 插槽。不建议在产品中使用，一般将其禁用，但管理员可使用它进行测试或者在信托网络中使用。默认端口为 16509。

默认传输，如果没有具体指定，则为 tls。

### 远程 URI

统一资源标识符 (URI) 是 virsh 和 libvirt 用来连接到远程主机的工具。URI 还可与带 --connect 参数的 virsh 命令一同使用执行单一命令或者在远程主机中迁移。

libvirt URI 使用通用格式 (方括号中的内容，即 "[ ]"，代表可选功能)：

```
driver[+transport]://[username@][hostname][:port]/[path][?extraparameters]
```

必须提供传输方法或者主机名以便使其与本地 URI 区分。

### 远程管理参数示例

- 在名为 towada 的主机中使用 SSH 传输和 SSH 用户名 ccurran 连接到远程 Xen 管理程序。

```
xen+ssh://ccurran@towada/
```

- 使用 TLS 在名为 towada 的主机中连接到远程 Xen 监控程序。

```
xen://towada/
```

- 使用 TLS 在名为 towada 的主机中连接到远程 Xen 监控程序。`no_verify=1` 告诉 libvirt 不要验证服务器证书。

```
xen://towada/?no_verify=1
```

- 使用 SSH 在主机 towada 中连接到远程 KVM 管理程序。

```
qemu+ssh://towada/system
```

### 测试示例

- 使用非标准 UNIX 插槽连接到本地 KVM 管理程序。在此特别提供了到该 Unix 插槽的完整路径。

```
qemu+unix:///system?socket=/opt/libvirt/run/libvirt/libvirt-sock
```

- 使用未加密 TCP/IP 将 libvirt 守护进程连接到端口 5000 中 IP 地址为 10.1.1.10 的服务器。这使用带默认设置的测试驱动程序。

```
test+tcp://10.1.1.10:5000/default
```

### 额外 URI 参数

在远程 URI 中可附加额外参数。下面的表格 13.1 “URI” 包含可识别的参数。其它参数将被忽略。注意：参数值必须是 URI 以外的值（就是说问号 (?) 是添加在参数前，特殊符号将转换成 URI 格式）。

名称	传输模式	Description	示例用法
name	所有模式	为 virConnectOpen 功能提供的名称。该名称通常由删除的传输、主机名、端口号、用户名以及远程 URI 中的附加参数组成，但在这个非常复杂的例子中，最好提供单独的名称。	name=qemu:///system
命令	ssh 和 ext	外部命令。ext 传输需要外部命令。ssh 默认为 ssh。PATH 是为该命令进行搜索。	command=/opt/openssh/bin/ssh
插槽	unix 和 ssh	到 UNIX 域套接字的路径将覆盖默认路径。在 ssh 传输中会将其传递给远程 netcat 命令（参见 netcat）。	socket=/opt/libvirt/run/libvirt/libvirt-sock
netcat	ssh	远程机器中 netcat 命令的名称。默认为 nc。	netcat=/opt/netcat/bin/nc

名称	传输模式	Description	示例用法
		在 ssh 传输中，libvirt 组成的 ssh 命令类似： <code>command -p port [-l username] hostname netcat -U socket</code> ，其中 <code>port</code> 、 <code>username</code> 、 <code>hostname</code> 可作为远程 URI 的一部分指定， <code>command</code> 、 <code>netcat</code> 和 <code>socket</code> 来自外部参数（或者合理的默认设置）。	
<code>no_verify</code>	<code>tls</code>	如果设定一个非零值，则会禁止客户端坚持服务器证书。注意：要禁用服务器检查客户端证书或者 IP 地址，您必须更改 <code>libvirtd</code> 配置。	<code>no_verify=1</code>
<code>no_tty</code>	<code>ssh</code>	如果设定一个非零值，则会在无法自动登录到远程机器时禁止 <code>ssh</code> 询问密码（在使用 <code>ssh-agent</code> 或者类似程序的情况下）。在您无法访问终端时使用这个值 -- 例如在使用 <code>libvirt</code> 的图形程序中。	<code>no_tty=1</code>

表 13.1.0 额外 URI 参数





---

## 部分 IV. 虚拟化参考指南

### 虚拟化命令、系统工具、 应用程序以及附加系统参考

这些章节提供 Fedora 中包含的虚拟化命令、系统工具以及应用程序的具体论述。这些章节是为需要高级功能性以及其他特性的用户准备的。

---

---

## 虚拟化工具

以下是在运行 Xen 的系统中有用的虚拟化管理工具和调试及联网工具。

### 系统管理工具

- vmstat
- iostat
- lsof

```
# lsof -i :5900
xen-vncfb 10635 root 5u IPv4 218738 TCP grumble.boston.redhat.com:5900 (LISTEN)
```

- qemu-img

### 高级调试工具

- systemTap
- crash
- xen-gdbserver
- sysrq
- sysrq t
- sysrq w
- sysrq c

### 联网

#### brctl

```
# brctl show
bridge name bridge id      STP enabled  interfaces
xenbr0      8000,fefffffffeff no           vif13,0
                                pdummy0
                                vif0,0
```

```
# brctl showmacs xenbr0
port no mac addr          is local?  aging timer
1   fe:ff:ff:ff:ff:ff     yes        0.00
```

```
# brctl showstp xenbr0
xenbr0
bridge id      8000,fefffffffeff
designated root 8000,fefffffffeff
root port      0                path cost        0
max age        20.00           bridge max age   20.00
hello time     2.00            bridge hello time 2.00
```

```
forward delay    0.00          bridge forward delay  0.00
aging time       300.01
hello timer      1.43          tcn timer              0.00
topology change timer 0.00          gc timer              0.02
flags

vif13.0 (3)
port id          8003          state                  forwarding
designated root   8000.fefffffffffff path cost              100
designated bridge 8000.fefffffffffff message age timer     0.00
designated port   8003          forward delay timer   0.00
designated cost   0            hold timer            0.43
flags

pdummy0 (2)
port id          8002          state                  forwarding
designated root   8000.fefffffffffff path cost              100
designated bridge 8000.fefffffffffff message age timer     0.00
designated port   8002          forward delay timer   0.00
designated cost   0            hold timer            0.43
flags

vif0.0 (1)
port id          8001          state                  forwarding
designated root   8000.fefffffffffff path cost              100
designated bridge 8000.fefffffffffff message age timer     0.00
designated port   8001          forward delay timer   0.00
designated cost   0            hold timer            0.43
flags
```

- ifconfig
- tcpdump

#### KVM 工具

- ps
- pstree
- top
- kvmtrace
- kvm\_stat

#### Xen 工具

- xentop
- xm dmesg
- xm log

## 使用 virsh 管理客户端

virsh 是用来管理客户端及其管理程序的命令行界面工具。

virsh 工具是构建在 libvirt 管理 API 上，可作为 `xm` 命令和图形客户端管理程序（`virt-manager`）的替代工具使用。非特权用户只能以只读模式使用 virsh。您可使用 virsh 为客户端机器执行脚本。

### virsh 命令快速参考

下表提供所有 virsh 命令行选项的快速参考。

命令	Description
help	打印基本帮助信息。
list	列出所有客户端。
dumpxml	输出客户端 XML 配置文件。
create	从 XML 配置文件生成客户端并启动新客户端。
start	启动未激活的客户端。
destroy	强制客户端停止。
define	为客户端输出 XML 配置文件。
domid	显示客户端 ID。
domuuid	显示客户端 UUID。
dominfo	显示客户端信息。
domname	显示客户端名称。
domstate	显示客户端状态。
quit	退出这个互动终端。
reboot	重新启动客户端。
restore	恢复以前保存在文件中的客户端。
resume	恢复暂停的客户端。
save	将客户端当前状态保存到某个文件中。
shutdown	关闭某个域。
suspend	暂停客户端。
undefine	删除与客户端关联的所有文件。
migrate	将客户端迁移到另一台主机中。

表015.1.0客户端管理命令

使用以下 virsh 命令管理客户端及管理程序资源：

命令	Description
setmem	为客户端设定分配的内存。
setmaxmem	为管理程序设定内存上限。
setvcpus	修改为客户端分配的虚拟 CPU 数目。
vcpuinfo	显示客户端的虚拟 CPU 信息。
vcupin	控制客户端的虚拟 CPU 亲和性。

命令	Description
domblockstat	显示正在运行的客户端的块设备统计。
domifstat	显示正在运行的客户端的网络接口统计。
attach-device	使用 XML 文件中的设备定义在客户端中添加设备。
attach-disk	在客户端中附加新磁盘设备。
attach-interface	在客户端中附加新网络接口。
detach-device	从客户端中分离设备，使用同样的 XML 描述作为命令 attach-device。
detach-disk	从客户端中分离磁盘设备。
detach-interface	从客户端中分离网络接口。

表 15.2 资源管理选项

这些是其它 virsh 选项：

命令	Description
version	显示 virsh 版本
nodeinfo	有关管理程序的输出信息

表 15.3 其它选项

### 连接至管理程序

使用 virsh 连接到管理程序会话：

```
# virsh connect {hostname OR URL}
```

其中 <name> 是管理程序所在的机器名。要初始化只读连接，请在上面的命令中附加 readonly。

### 创建虚拟机 XML 转储（配置文件）

使用 virsh 输出客户端 XML 配置文件：

```
# virsh dumpxml {domain-id, domain-name or domain-uuid}
```

这个命令将客户端 XML 配置文件输出为标准输出（stdout）。您可以通过将输出导入文件来保存数据。以下为将输出导入名为 *guest.xml* 的文件示例：

```
# virsh dumpxml GuestID > guest.xml
```

这个文件 *guest.xml* 可重新生成客户端（请参考 [00000000](#)）。您可以编辑这个 XML 配置文件来配置额外的设备或者部署额外的客户端。有关修改 virsh dumpxml 所生成文件的详情请参考 [18.1](#) “[virsh](#) XML [0000](#)”。

virsh dumpxml 输出示例：

```
# virsh dumpxml r5b2-mySQL01
<domain type='xen' id='13'>
```

```

<name>r5b2-mysql01</name>
<uuid>4a4c59a7ee3fc78196e4288f2862f011</uuid>
<bootloader>/usr/bin/pygrub</bootloader>
<os>
  <type>linux</type>
  <kernel>/var/lib/libvirt/vmlinuz.2dgnU_</kernel>
  <initrd>/var/lib/libvirt/initrd.UQafMw</initrd>
  <cmdline>ro root=/dev/VolGroup00/LogVol00 rhgb quiet</cmdline>
</os>
<memory>512000</memory>
<vcpu>1</vcpu>
<on_poweroff>destroy</on_poweroff>
<on_reboot>restart</on_reboot>
<on_crash>restart</on_crash>
<devices>
  <interface type='bridge'>
    <source bridge='xenbr0' />
    <mac address='00:16:3e:49:1d:11' />
    <script path='vif-bridge' />
  </interface>
  <graphics type='vnc' port='5900' />
  <console tty='/dev/pts/4' />
</devices>
</domain>

```

### 使用配置文件创建客户端

可使用 XML 配置文件创建客户端。您可从之前生成的客户端中复制现有 XML，也可以使用 `dumpxml` 选项（请参考 [XML](#)）。使用 `virsh` 创建客户端：

```
# virsh create configuration_file.xml
```

### 编辑客户端配置文件

除了使用 `dumpxml` 选项（请参考 [XML](#)），还可在其运行时或者离线时编辑客户端。`virsh edit` 命令有这个功能。例如：编辑名为 `softwaretesting` 的客户端：

```
# virsh edit softwaretesting
```

这样可打开文本编辑器。默认文本编辑器为 `$EDITOR` shell 参数（默认将其设定为 `vi`）。

### 挂起客户端

使用 `virsh` 挂起客户端：

```
# virsh suspend {domain-id, domain-name or domain-uuid}
```

当客户端处于挂起状态时，它仍旧消耗系统内存但不消耗处理器资源。客户端挂起时不会有磁盘或网络 I/O。这个操作是立即生效的，必须使用 `resume` 选项（[XML](#)）才能重新启动客户端。

### 恢复客户端

使用 virsh 的 resume 选项恢复客户端：

```
# virsh resume {domain-id, domain-name or domain-uuid}
```

这个操作是立即生效的，客户端状态将处于 suspend 和 resume 的循环中。

### 保存客户端

使用 virsh 命令将客户端的当前状态保存到文件中：

```
# virsh save {domain-name, domain-id or domain-uuid} filename
```

这个命令停止您指定的客户端并将数据保存到文件中，整个过程所需的时间根据您为客户端分配的内存数量而有所不同。您可以用 restore 选项 (00000) 来恢复客户端状态。保存与暂停类似，它除了暂停客户端外还保存了客户端的当前状态。

### 恢复客户端

请使用 virsh 恢复之前使用 virsh save 命令保存的客户端 (00000)：

```
# virsh restore filename
```

这个命令重新启动了保存的客户端，这会需要一段时间。客户端名称和 UUID 都会被保留，但会分配一个新的 id。

### 关闭客户端

使用 virsh 命令关闭客户端：

```
# virsh shutdown {domain-id, domain-name or domain-uuid}
```

修改客户端配置文件中的 on\_shutdown 参数可控制重启客户端的行为。

### 重新启动客户端

使用 virsh 重启客户端：

```
#virsh reboot {domain-id, domain-name or domain-uuid}
```

修改客户端配置文件中的 on\_reboot 参数控制重启客户端的行为。

### 强制客户端停止

使用 virsh 强制客户端停止：

```
# virsh destroy {domain-id, domain-name or domain-uuid}
```



---

这个命令执行一个立刻关闭动作并停止指定的客户端。使用 `virsh destroy` 命令可破坏客户端文件系统。客户端无法响应时请只使用 `destroy` 选项。对于半虚拟客户端，请使用 `shutdown` 选项（[\[1\]](#)）。

### 获得客户端域 ID

要获得客户端域 ID：

```
# virsh domid {domain-name or domain-uuid}
```

### 获得客户端域名称

要获得客户端域名称：

```
# virsh domname {domain-id or domain-uuid}
```

### 获得客户端 UUID

要获得客户端全局唯一识别符号（UUID）：

```
# virsh domuuid {domain-id or domain-name}
```

virsh domuuid 命令输出示例：

```
# virsh domuuid r5b2-mysQL01
4a4c59a7-ee3f-c781-96e4-288f2862f011
```

### 显示客户端信息

使用带客户端域 ID、域名或者 UUID 的 virsh 命令可显示指定客户端的信息：

```
# virsh dominfo {domain-id, domain-name or domain-uuid}
```

以下是 virsh dominfo 命令的输出示例：

```
# virsh dominfo r5b2-mysQL01
id:          13
name:        r5b2-mysql01
uuid:        4a4c59a7-ee3f-c781-96e4-288f2862f011
os type:     linux
state:       blocked
cpu(s):      1
cpu time:    11.0s
max memory:  512000 kb
used memory: 512000 kb
```

### 显示主机信息

要显示主机信息：

```
# virsh nodeinfo
```

virsh nodeinfo 命令输出示例：

```
# virsh nodeinfo
CPU model          x86_64
CPU (s)            8
CPU frequency      2895 Mhz
CPU socket(s)      2
Core(s) per socket 2
Threads per core:  2
Numa cell(s)       1
Memory size:       1046528 kb
```

这显示了节点信息和支持虚拟化进程的机器。

### 显示客户端

使用 virsh 显示客户端列表以及当前状态：

```
# virsh list
```

其它可用选项包括：

--inactive 选项列出不活动的域（已经被定义但目前不活动的域），以及

--all 选项列出所有客户端。例如：

```
# virsh list --all
Id Name          State
-----
0 Domain-0       running
1 Domain202      paused
2 Domain010      inactive
3 Domain9600     crashed
```

将 virsh list 输出结果分类成以下 6 个状态之一（如下）。

- **running** 状态是指目前在 CPU 中活跃的客户端。
- 列为 **blocked** 的客户端是被阻断的，且目前没有运行或者无法运行。这是由于客户端正在等待 I/O（传统的等待状态）或者客户端处于睡眠模式。
- **paused** 状态列出暂停的域。如果管理员在 `virt-manager`、`xm pause` 或者 `virsh suspend` 中使用 `pause` 标记就会出现这种状态。当客户端处于暂停状态时，它仍会消耗内存和其它资源，但无法从管理程序中调度 CPU 资源。
- **shutdown** 状态是用于处于关闭过程中的客户端。向该客户端发送关闭信号并使其温和地停止操作。这可能不适用于所有客户端操作系统，有些操作系统对这些信号没有响应。
- 处于 **dying** 状态的域是处于濒死状态，即该域还没有完全关闭或者崩溃。

- 
- 处于 `crashed` 状态的客户端是在运行时失败且无法再运行。这个状态只在将客户端配置为崩溃时不重启时出现。

### 显示虚拟 CPU 信息

使用 `virsh` 显示客户端中虚拟 CPU 的信息：

```
# virsh vcpuinfo {domain-id, domain-name or domain-uuid}
```

`virsh vcpuinfo` 命令输出示例：

```
# virsh vcpuinfo r5b2-mysql01
VCPU:    0
CPU:     0
State:   blocked
CPU time: 0.0s
CPU Affinity: yy
```

### 配置虚拟 CPU 亲和性

使用物理 CPU 配置虚拟 CPU 的亲和性：

```
# virsh vcpupin {domain-id, domain-name or domain-uuid} vcpu, cpulist
```

其中 `vcpu` 是虚拟 VCPU 号而 `cpulist` 列出了 CPU 的物理序号。

### 配置虚拟 CPU 计数

使用 `virsh` 修改分配给客户端的 CPU 数：

```
# virsh setvcpus {domain-name, domain-id or domain-uuid} count
```

新 `count` 值不能超过创建客户端时指定的数目。

### 配置内存分配

使用 `virsh` 修改客户端内存分配：

```
# virsh setmem {domain-id or domain-name} count
```

您必须以 KB 为单位指定 `count`。新 `count` 值不能超过您创建客户端时指定的数值。大多数客户端操作系统需要不低于 64MB 的值。较高的最大内存值不会影响活跃的客户端，除非新值较小，这会降低可用内存量。

### 显示客户端块设备信息

使用 `virsh domblkstat` 显示运行的客户端的块设备统计。

```
# virsh domblkstat GuestName block-device
```

### 显示客户端网络设备信息

virsh domifstat 为运行的客户端显示网络接口统计。

```
# virsh domifstat GuestName interface-device
```

### 使用 virsh 迁移客户端

可使用 virsh 将某个客户端迁移到另一台主机中。将域迁移到另一台主机中。添加 --live 进行实时迁移。migrate 命令接受以下格式的参数：

```
# virsh migrate --live GuestName DestinationURL
```

--live 参数是可选的。实时迁移需要添加 --live 参数。

The *GuestName* parameter represents the name of the guest which you want to migrate.

The *DestinationURL* parameter is the URL or hostname of the destination system. The destination system must run the same version of Fedora, be using the same hypervisor and have libvirt running.

Once the command is entered you will be prompted for the root password of the destination system.

### 管理虚拟网络

这部分包括使用 virsh 管理虚拟网络。要列出虚拟网络：

```
# virsh net-list
```

这个命令产生的输出类似如下：

```
# virsh net-list
Name      State  Autostart
-----
default   active yes
vnet1     active yes
vnet2     active yes
```

要查看某一特定虚拟网络的信息：

```
# virsh net-dumpxml NetworkName
```

这以 XML 格式显示指定虚拟网络的信息：

```
# virsh net-dumpxml vnet1
<network>
  <name>vnet1</name>
  <uuid>98361b46-1581-acb7-1643-85a412626e70</uuid>
  <forward dev='eth0' />
  <bridge name='vnet0' stp='on' forwardDelay='0' />
  <ip address='192.168.100.1' netmask='255,255,255,0'>
```

---

```
<dhcp>
  <range start='192.168.100.128' end='192.168.100.254' />
</dhcp>
</ip>
</network>
```

其它用来管理虚拟网络的 virsh 命令：

- virsh net-autostart **network-name** — 自动启动指定为 **network-name** 的网络。
- virsh net-create **XMLfile** — 使用现有 XML 文件创建并启动新网络。
- virsh net-define **XMLfile** — 使用现有 XML 文件创建新网络设备但不启动。
- virsh net-destroy **network-name** — 销毁指定为 **network-name** 的网络。
- virsh net-name **networkUUID** — 将指定的 **networkUUID** 转换为网络名称。
- virsh net-uuid **network-name** — 将指定的 **network-name** 转换为网络 UUID。
- virsh net-start **nameOfInactiveNetwork** — 启动不活跃的网络。
- virsh net-undefine **nameOfInactiveNetwork** — 删除不活跃网络的定义。



# 用虚拟机管理器（virt-manager）管理客户端

本部分内容论述了虚拟机管理器（virt-manager）窗口、对话框及各种的 GUI 控制。

virt-manager 提供管理程序以及您系统和远程机器中客户端的图形视图。您可使用 virt-manager 定义半虚拟和全虚拟客户端。virt-manager 可执行虚拟化管理任务，其中包括：

- 分配内存，
- 分配虚拟 CPU，
- 监控操作性能，
- 保存和恢复、暂停和继续、关闭和启动虚拟客户端，
- 链接到文本和图形控制台，以及
- 实时和离线迁移。

## 16.1.0 打开连接窗口

首先出现这个窗口，提示用户选择管理程序会话。非特权用户可启动一个只读会话。根用户可启动具有全部读/写状态的会话。一般使用请选择「本地 Xen 主机」选项或者 QEMU（在 KVM 中）。

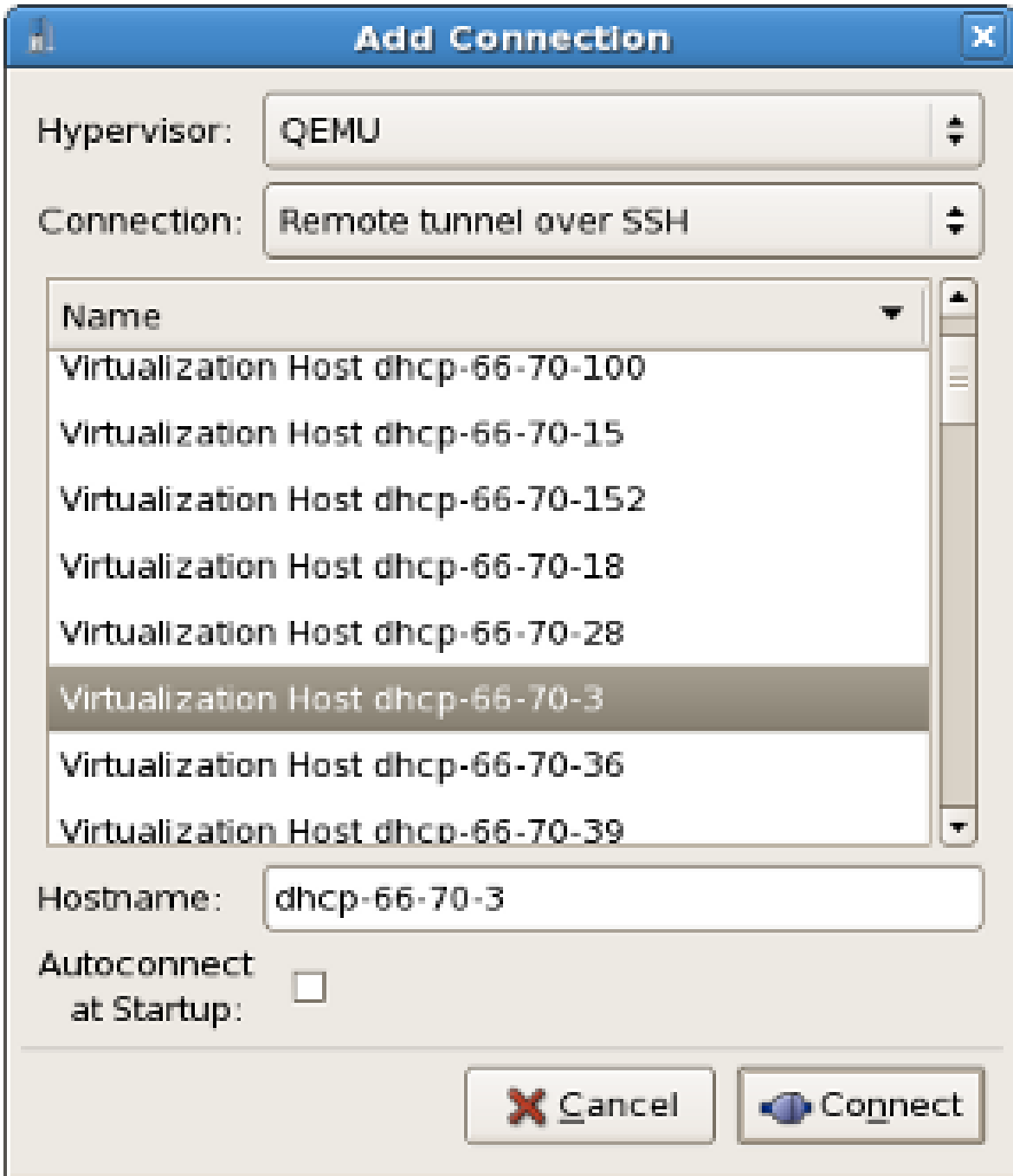


图16.1.0虚拟机管理器连接窗口

## 16.2.0虚拟机管理器主窗口

这个主窗口显示了所有运行的虚拟机及目前分配给它们的资源（包括 domain0），您可以决定显示哪些字段。双击某个虚拟机将出现这个特定机器的相应控制台。选择一个虚拟机并双击「详情」按钮显示那台机器的详情窗口。您还可以进入「文件」菜单创建一个新虚拟机。



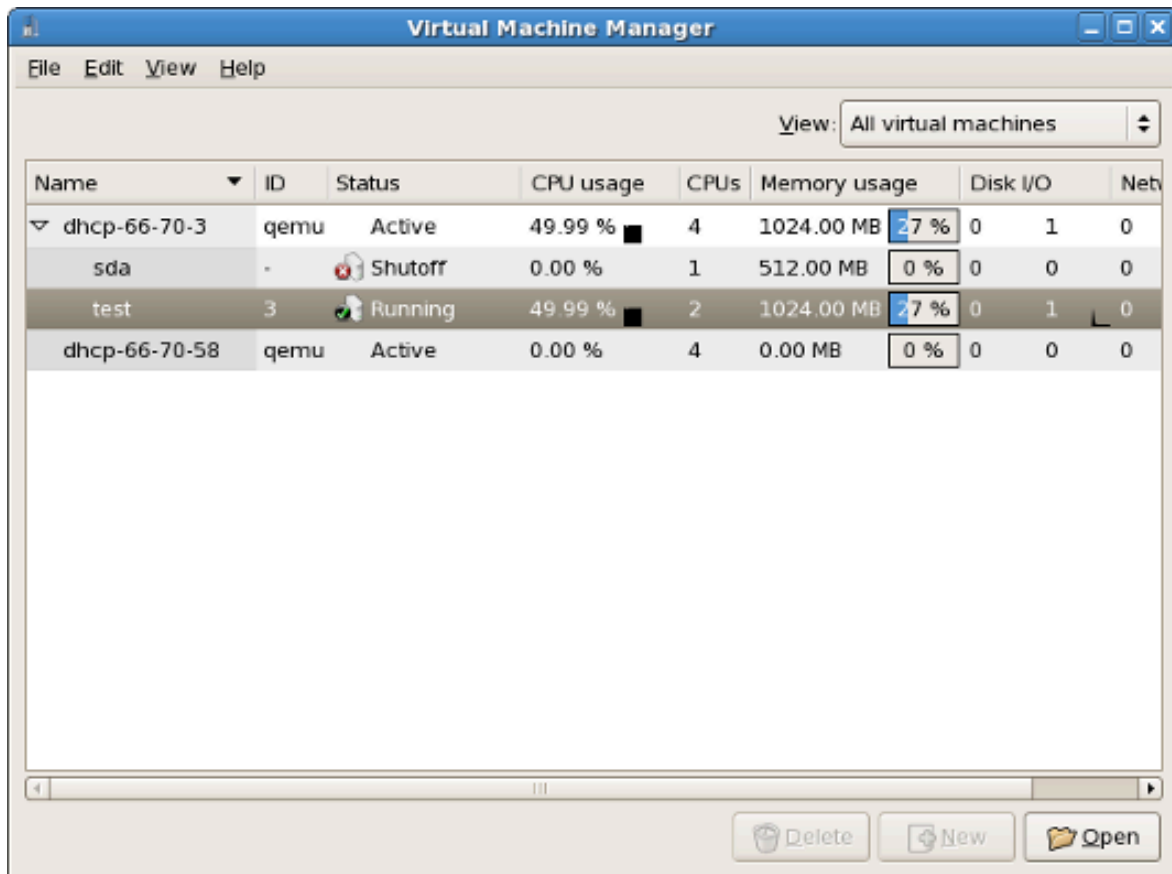


图16.2.0虚拟机管理器主窗口

### 16.3.0虚拟机管理器详情窗口

这个窗口显示 virt-manager 提供的客户端实时资源使用的图形和统计数据。UUID 字段显示了虚拟机的全局唯一标识符。

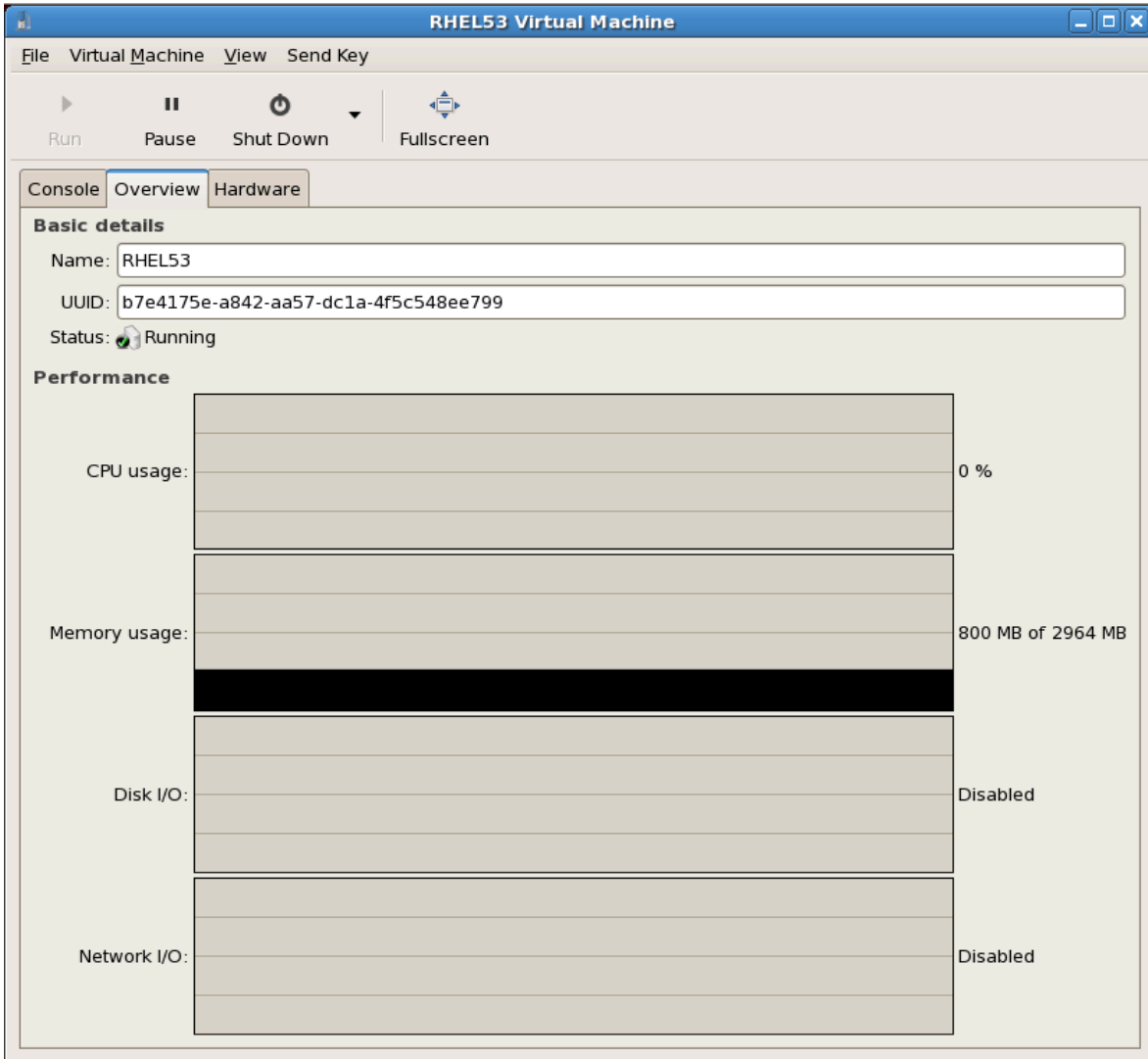


图16.3. virt-manager 详情窗口

## 16.4. 虚拟机图形控制台

这个窗口显示了虚拟机的图形控制台。半虚拟客户端和全虚拟客户端使用不同的技术导出其本地虚拟帧缓冲，但是这两种技术都使用 VNC 使其可用于虚拟机管理器的控制台窗口。如果您没有将虚拟机设置为需要验证，则在显示前虚拟机图形控制台会提示您输入密码。

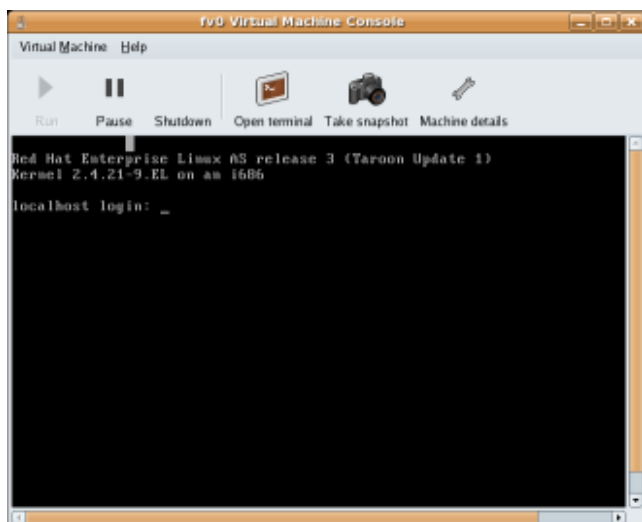


图 16.4. 图形控制台窗口



### 安全性和 VNC 注记

很多安全专家认为 VNC 不安全，但是我们做了一些修改以便在 Fedora 中安全使用 VNC 进行虚拟化。客户端机器只侦听本地主机（dom0）的回送地址（127.0.0.1）。这样就保证了只有主机中那些有 shell 特权的程序可以通过 VNC 访问 virt-manager 和虚拟机。

将根据 [13](#) 的说明执行远程管理。TLS 可提供管理客户端和主机系统的企业级安全性。

您的本地桌面可以截获组合键（如 Ctrl+Alt+F11）以防止其被发送到客户端机器。您可以使用 virt-manager 的 '粘滞键' 功能来发送这些键序列。您必须按任何修改键（如 Ctrl 或 Alt）三次，这个键才会被激活，直到按了下一个非修饰符键。您可以通过顺序按下 'Ctrl Ctrl Ctrl Alt+F11' 将 Ctrl-Alt-F11 发送给客户端。

## 16.5. Starting virt-manager

要启动 virt-manager 会话，请打开「应用程序」菜单，然后点击「系统工具」，并选择「虚拟机管理器」（virt-manager）。

此时会出现 virt-manager 主窗口。

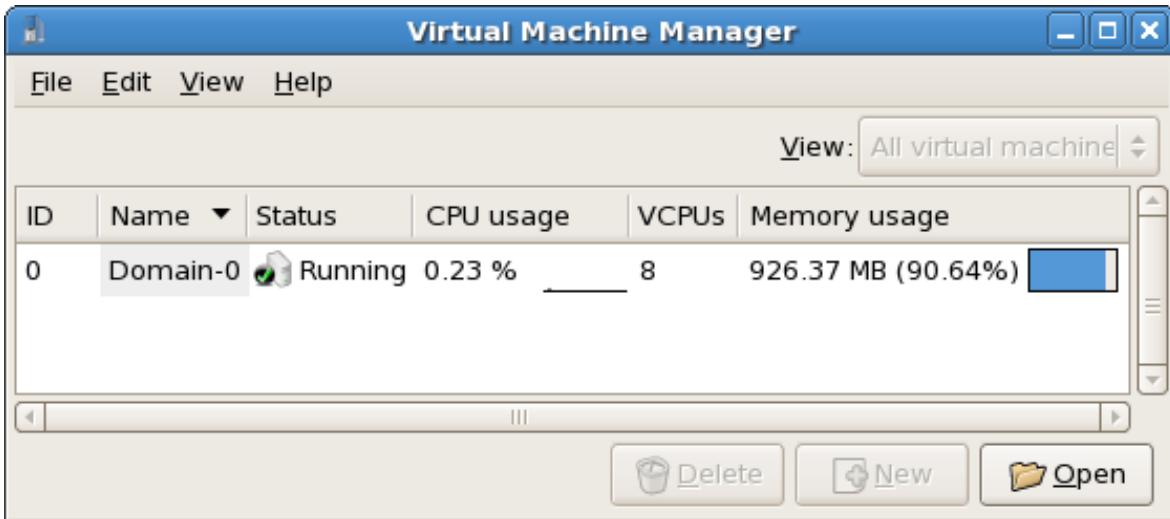


图16.5. 启动virt-manager

另外，您可以在下面命令中演示的 ssh 远程启动 virt-manager：

```
ssh -X host's address[remotehost]# virt-manager
```

使用 ssh 管理虚拟机和主机在 13.1 “SSL 配置”中有进一步的论述。

## 16.6. 恢复保存的机器

在启动虚拟机管理器后，主窗口中会显示系统中的所有虚拟机。Domain 0 是您的主机系统。如果没有出现任何机器，说明目前系统中没有虚拟机在运行。

恢复以前保存的会话：

1. 在「文件」菜单中，选择「恢复保存的机器」。

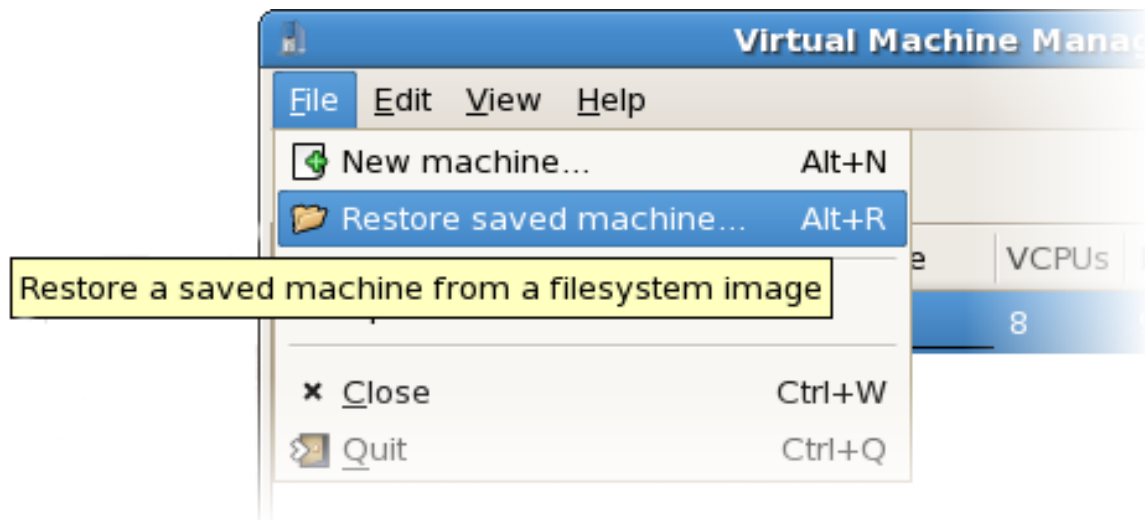


图16.6. 恢复虚拟机

2. 出现「恢复虚拟机」主窗口。

3. 导航至正确的目录并选择保存的会话文件。
4. 点击「打开」。

虚拟机管理器主窗口中会出现保存的虚拟系统。

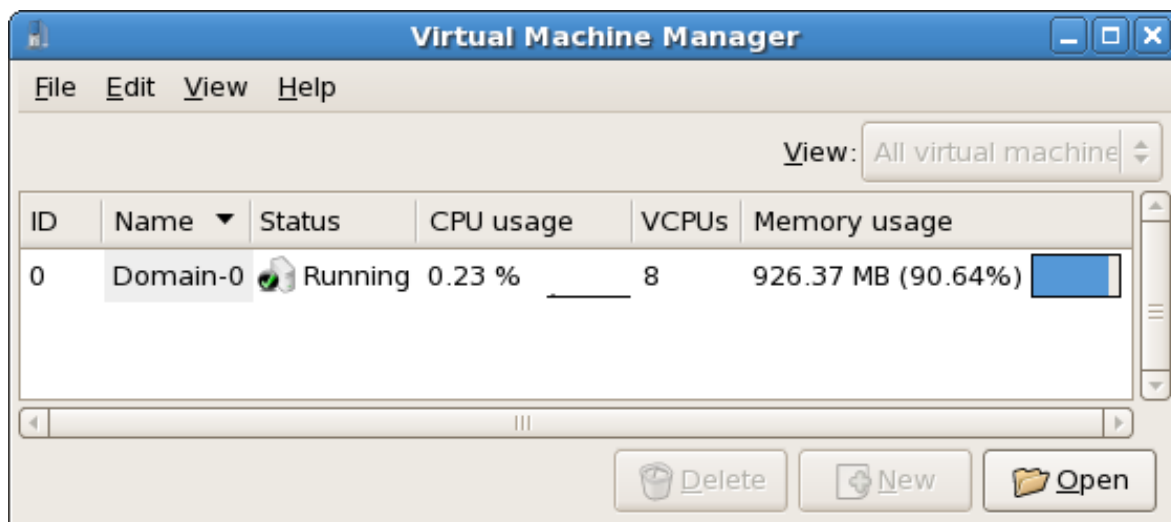


图16.7.0恢复的虚拟机管理器会话

## 16.7.0显示客户端详情

您可以使用虚拟机管理程序来查看系统中的任意虚拟机活动数据。

查看虚拟系统细节：

1. 在虚拟机管理器主窗口中选中您要查看的虚拟机。

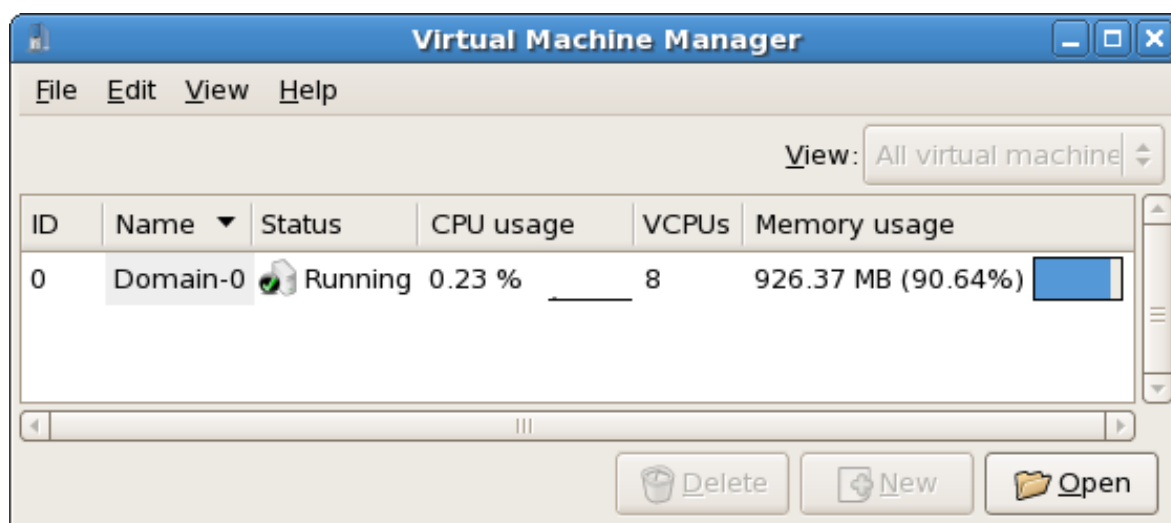


图16.8.0选择要显示的虚拟机

2. 在虚拟机管理器的「编辑」菜单中选择「机器详情」（或在虚拟机管理器主窗口的底部点击「详情」按钮）。

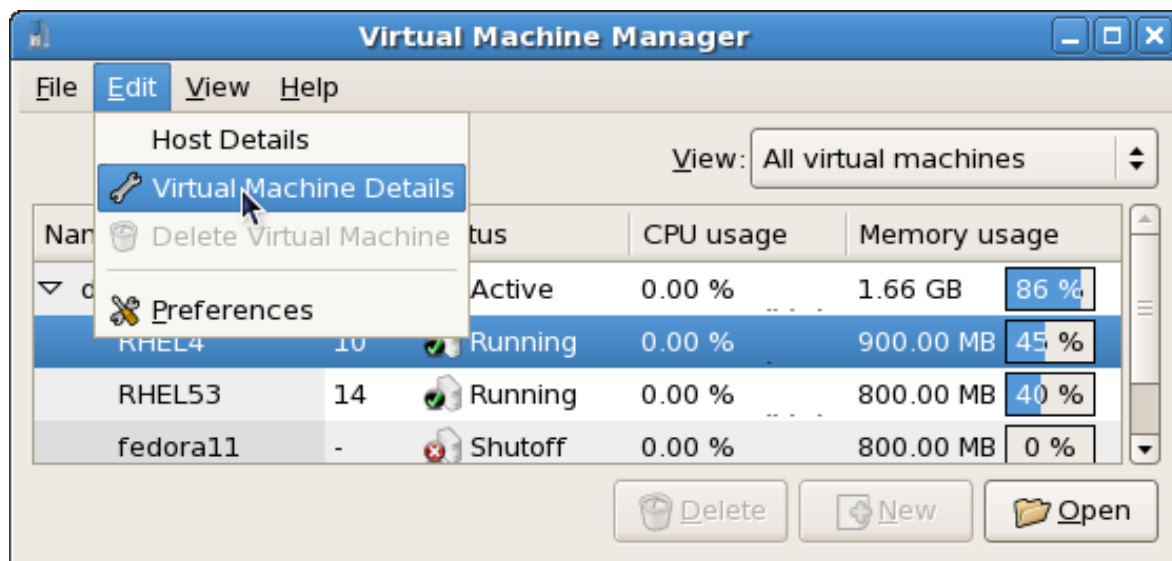


图16.9.0显示虚拟机详情菜单

出现虚拟机详情总览窗口。这个窗口显示了您指定域的 CPU 和内存使用情况。

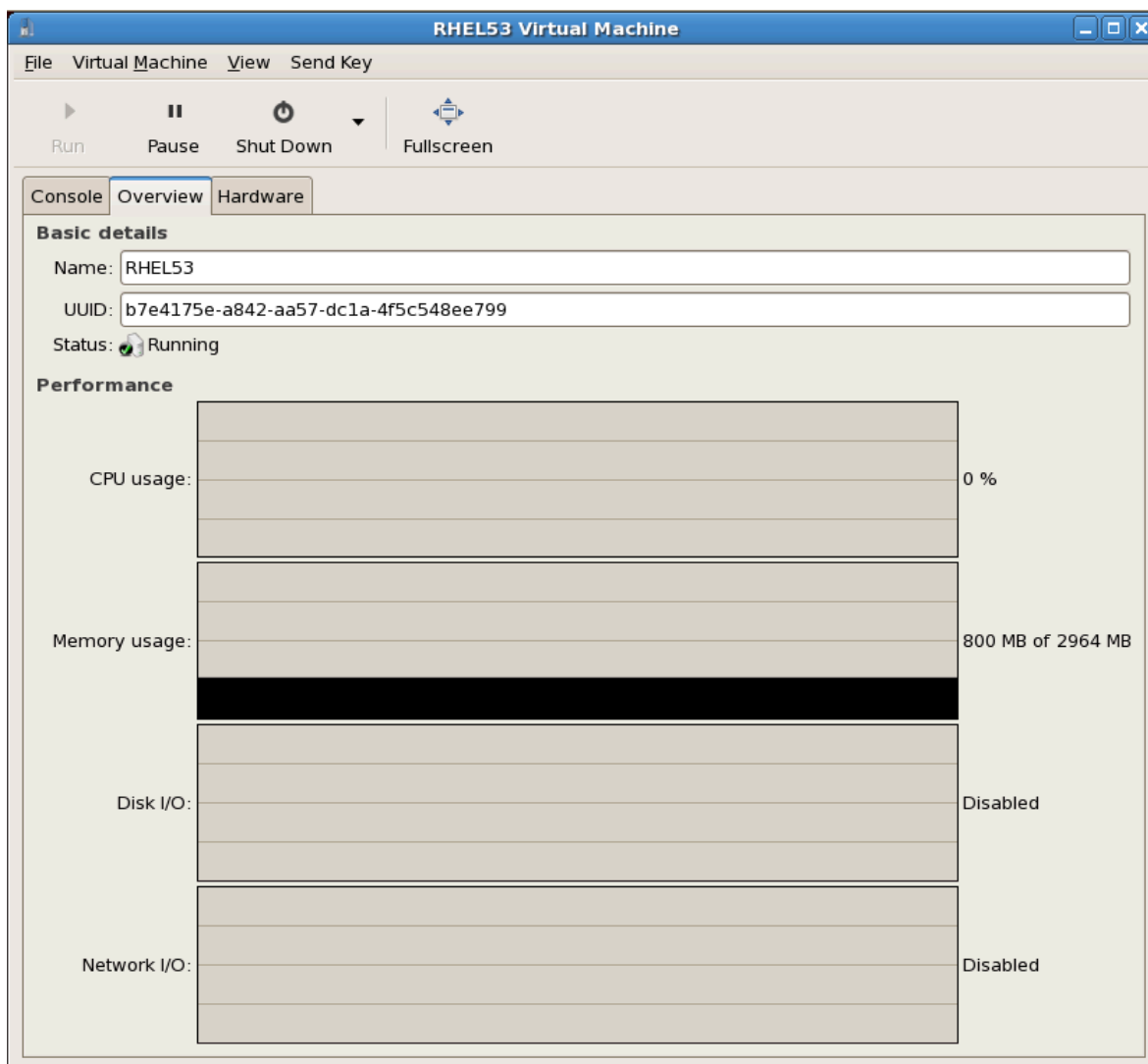


图016.10.0显示客户端详情总览

3. 在「虚拟机详情」窗口中点击「硬件」标签。  
出现「虚拟机详情硬件」窗口。

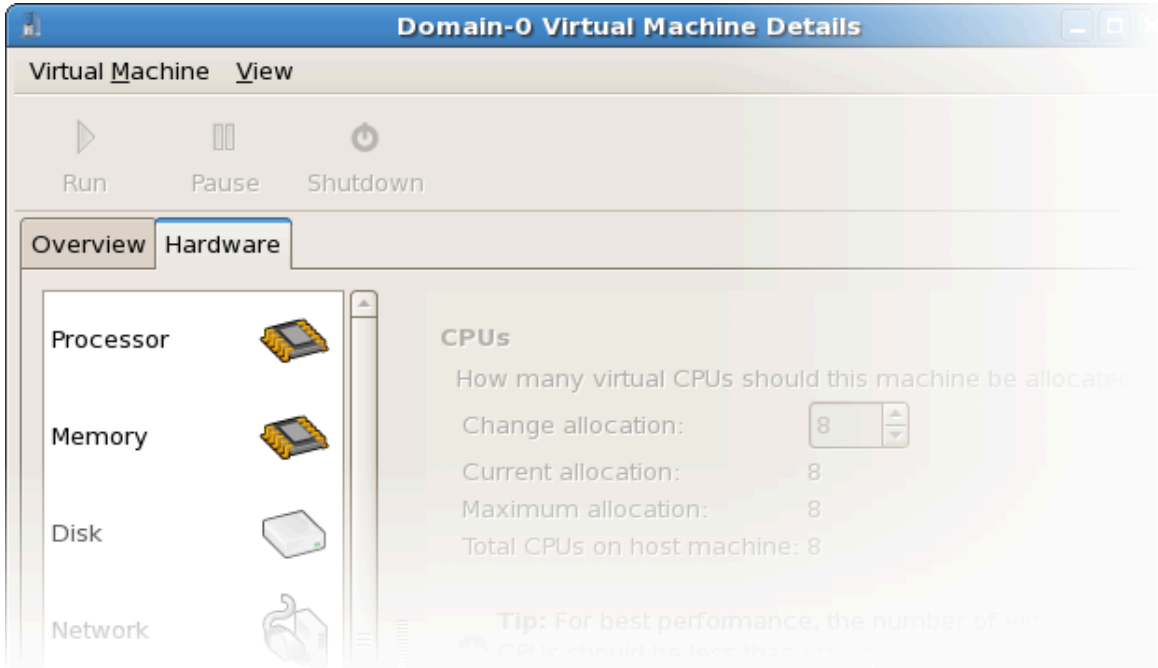


图016.11.0显示客户端硬件详情

4. 在「硬件」标签中，点击「处理器」查看或修改当前的处理器分配情况。

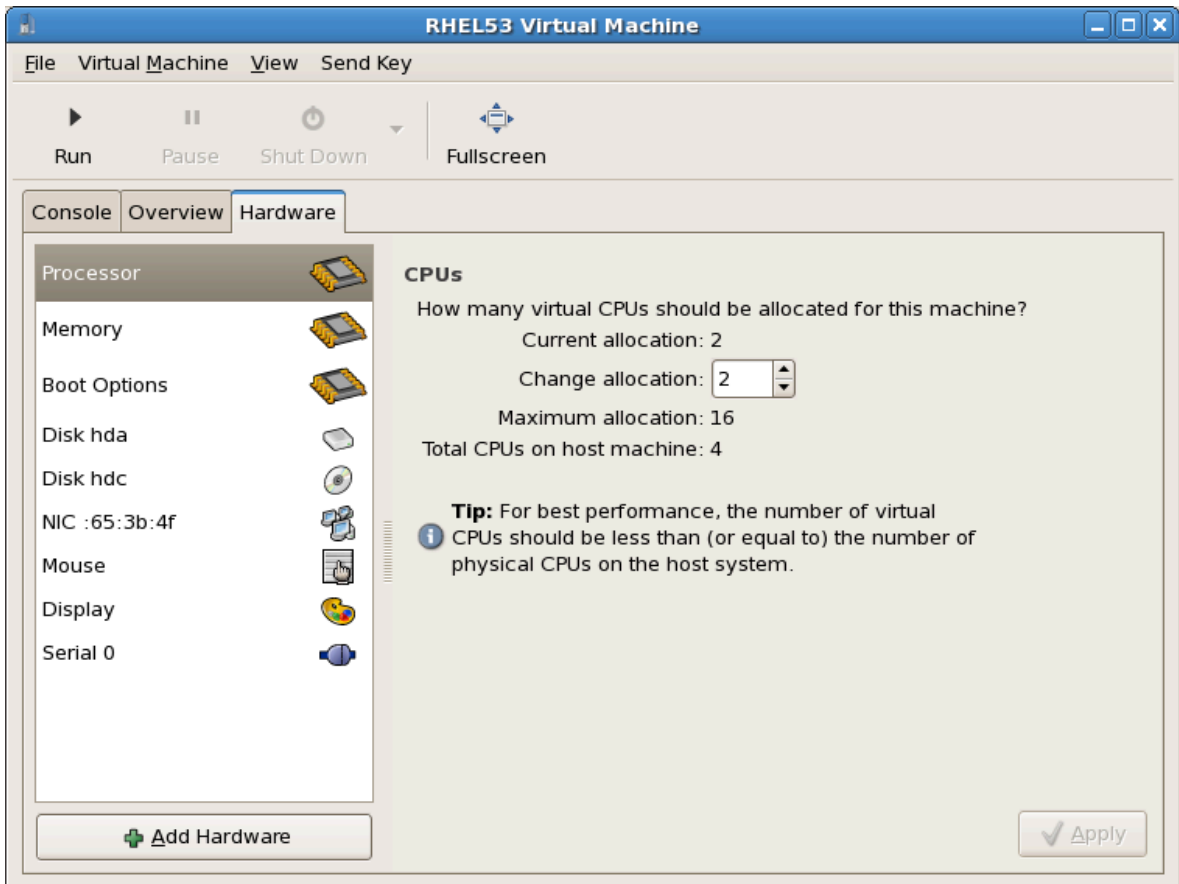


图016.12.0处理器分配面板



- 在「硬件」窗口，点击「内存」可以查看和修改当前的内存分配。

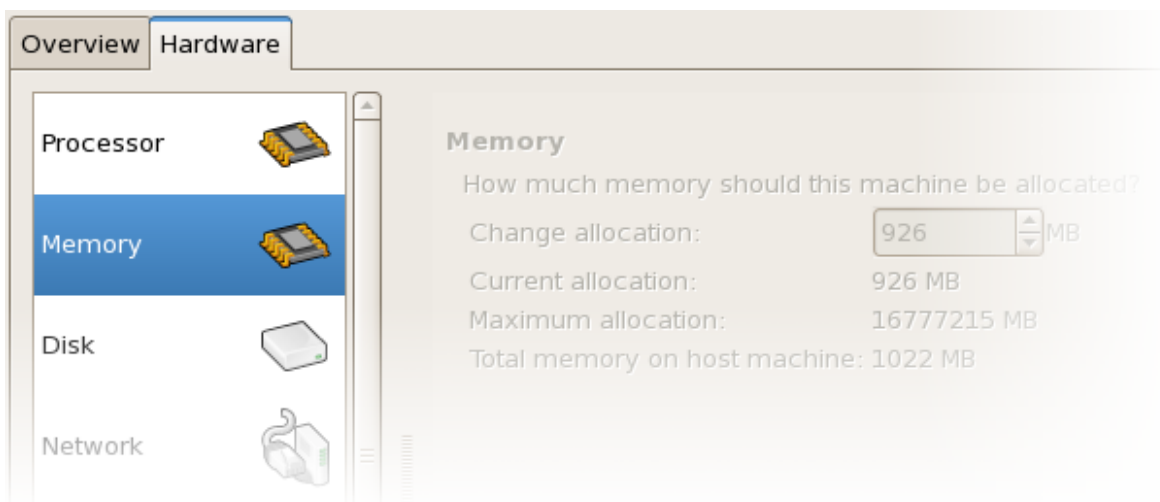


图016.13.0显示内存分配

- 在「硬件」标签中点击「磁盘」可以查看或修改当前的硬盘配置。

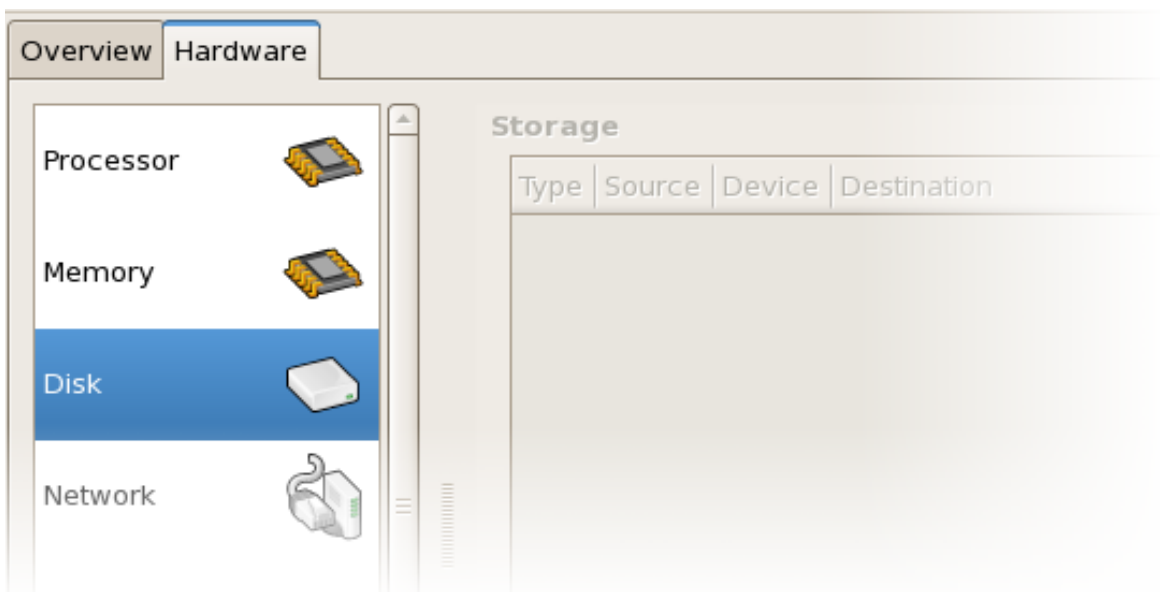


图016.14.0显示磁盘配置

- 在「硬件」标签中点击「网络」可以查看或修改当前的网络配置。

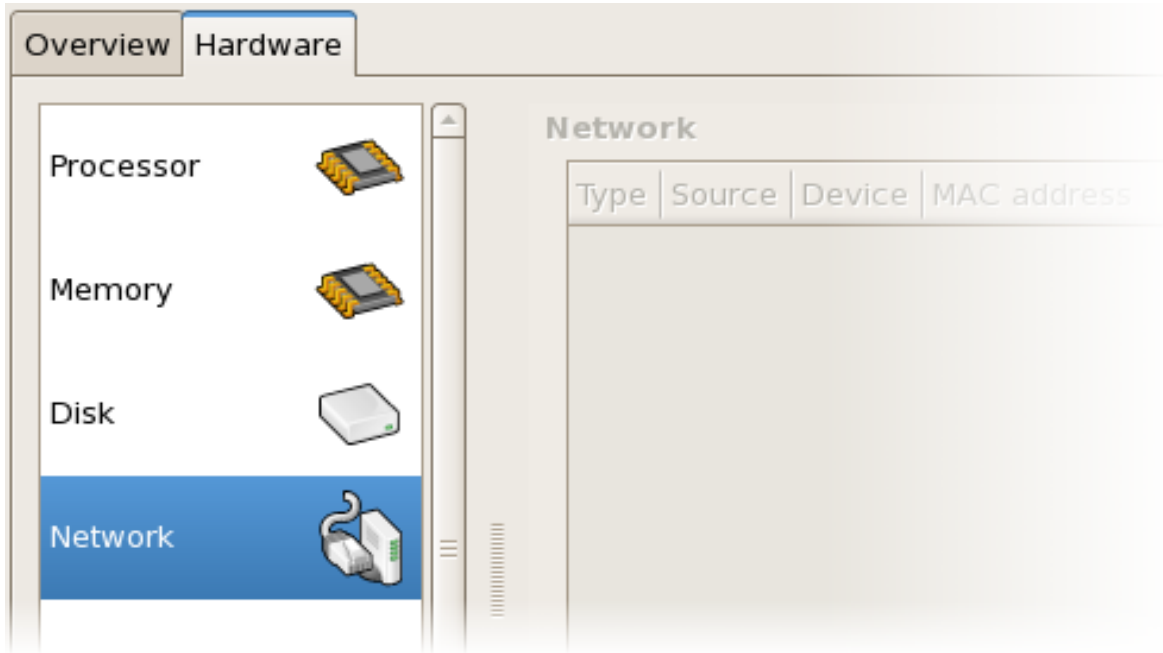


图16.15.0显示网络配置

## 16.8.0 状态监控

您可以使用虚拟机管理器来修改虚拟系统的状态监控。

配置状态监控并启用控制台：

1. 在「编辑」菜单中选择「首选项」。

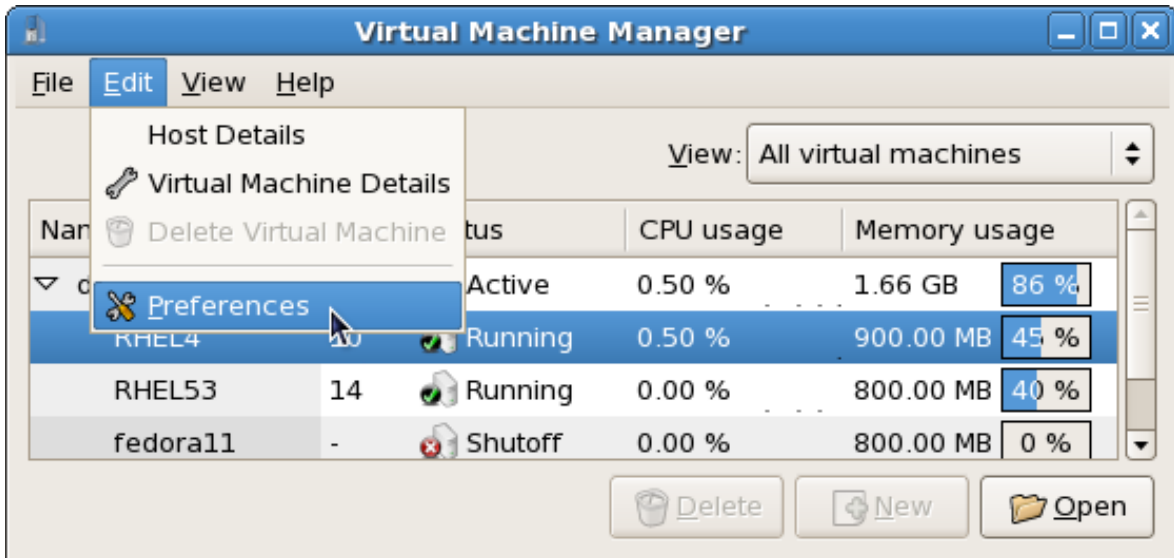


图16.16.0修改客户端首选项

出现虚拟机管理器首选项窗口。

2. 在状态监控区域选择框指定您希望系统更新的时间间隔（以秒为单位）。

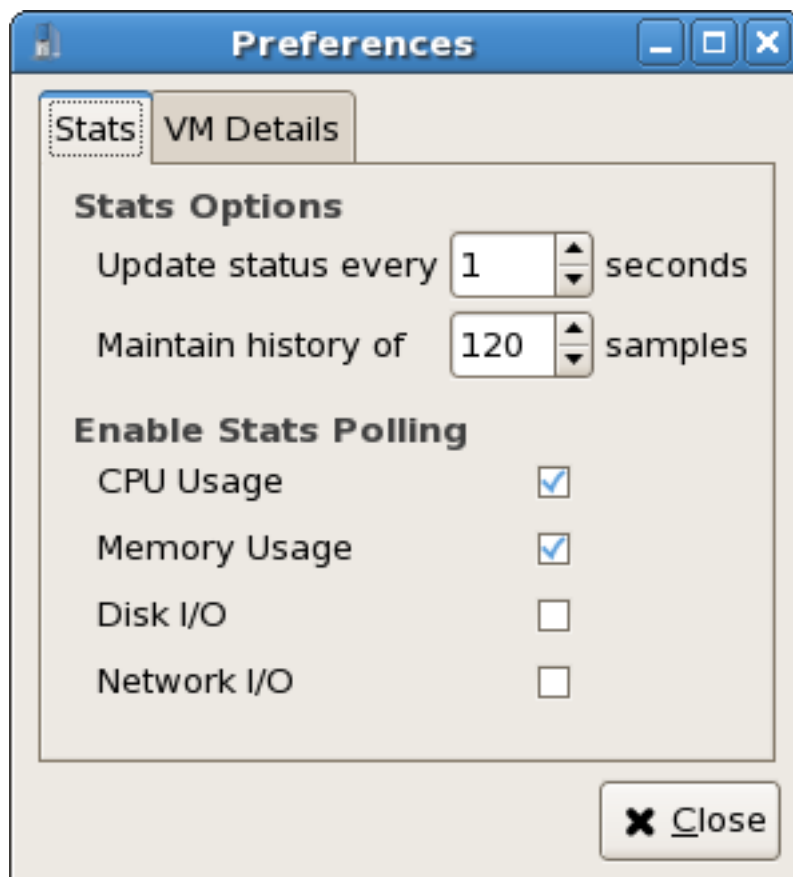


图 16.17.0 配置状态监控

3. 在控制台区，指定怎样打开控制台和输入设备。

## 16.9.0 显示客户端识别符

查看您系统中所有虚拟机的客户端 ID：

1. 在「查看」菜单中选择「域 ID」复选框。

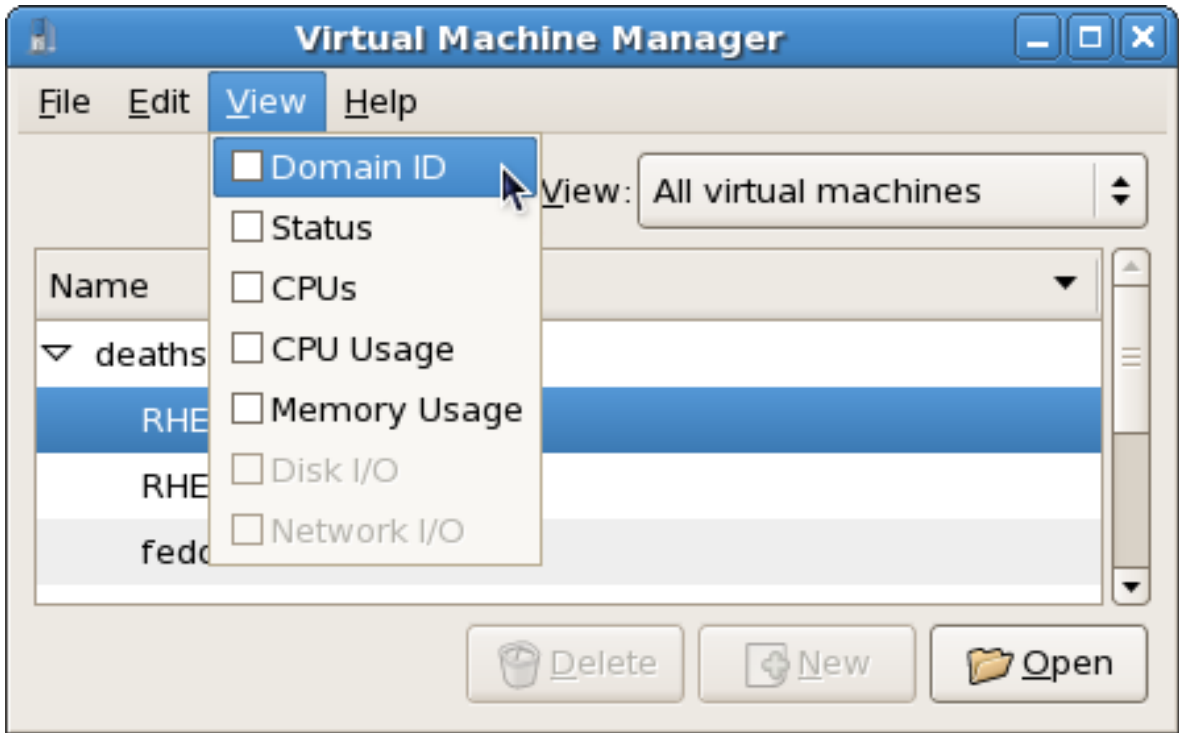


图16.18. 查看客户端 ID

- 虚拟机管理器列出了系统中所有域的域 ID。

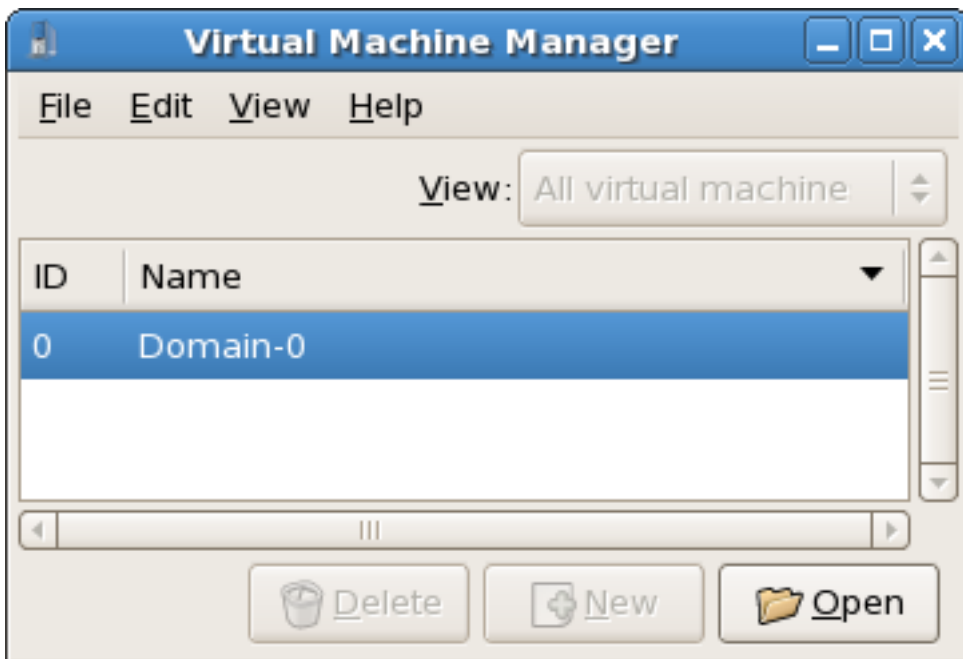


图16.19. 显示域 ID

## 16.10. 显示客户端状态

查看系统中所有虚拟机的状态：

- 在「查看」菜单中选择「状态」复选框。

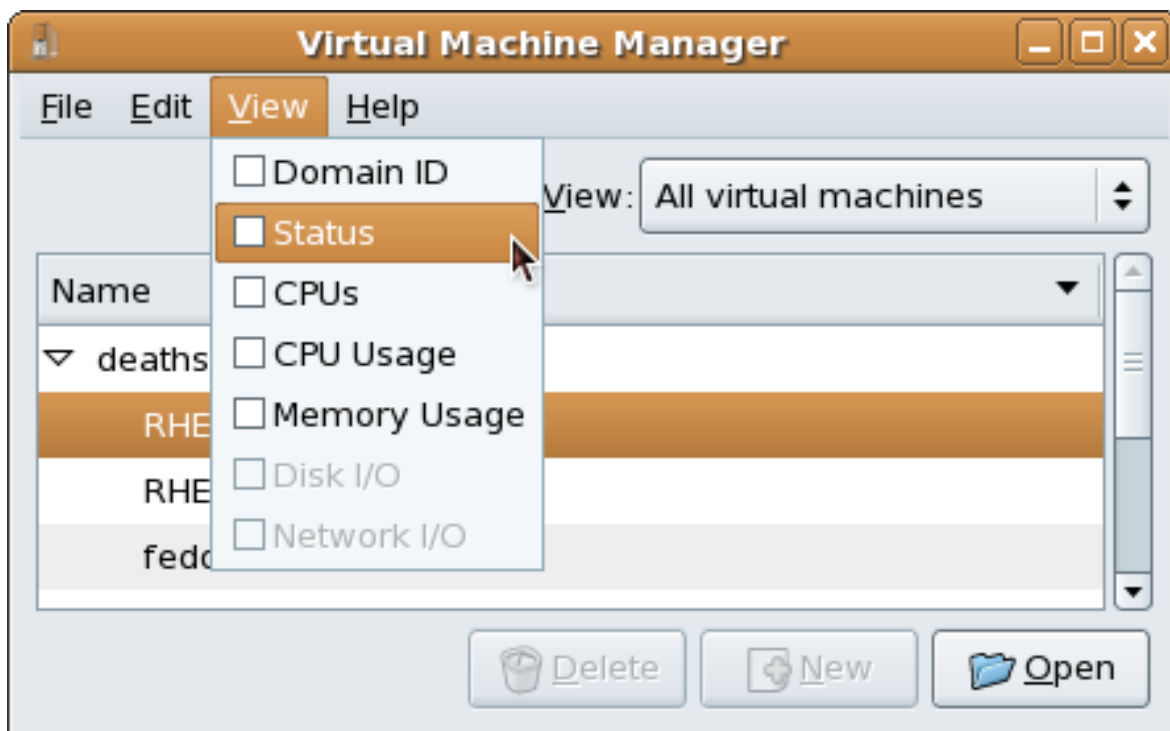


图 16.20. 选择虚拟机状态

- 虚拟机管理器列出了系统中所有虚拟机的状态。

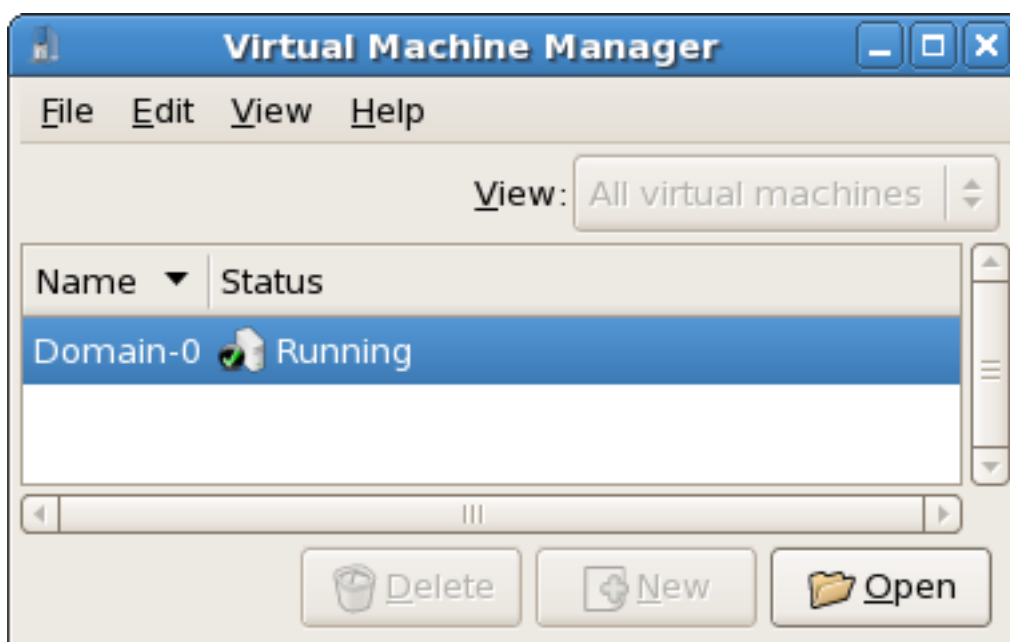


图 16.21. 显示虚拟机状态

## 16.11. 显示虚拟 CPU

查看系统中所有虚拟机的虚拟 CPU 的数量：

- 在「查看」菜单中选择「虚拟 CPU」复选框。

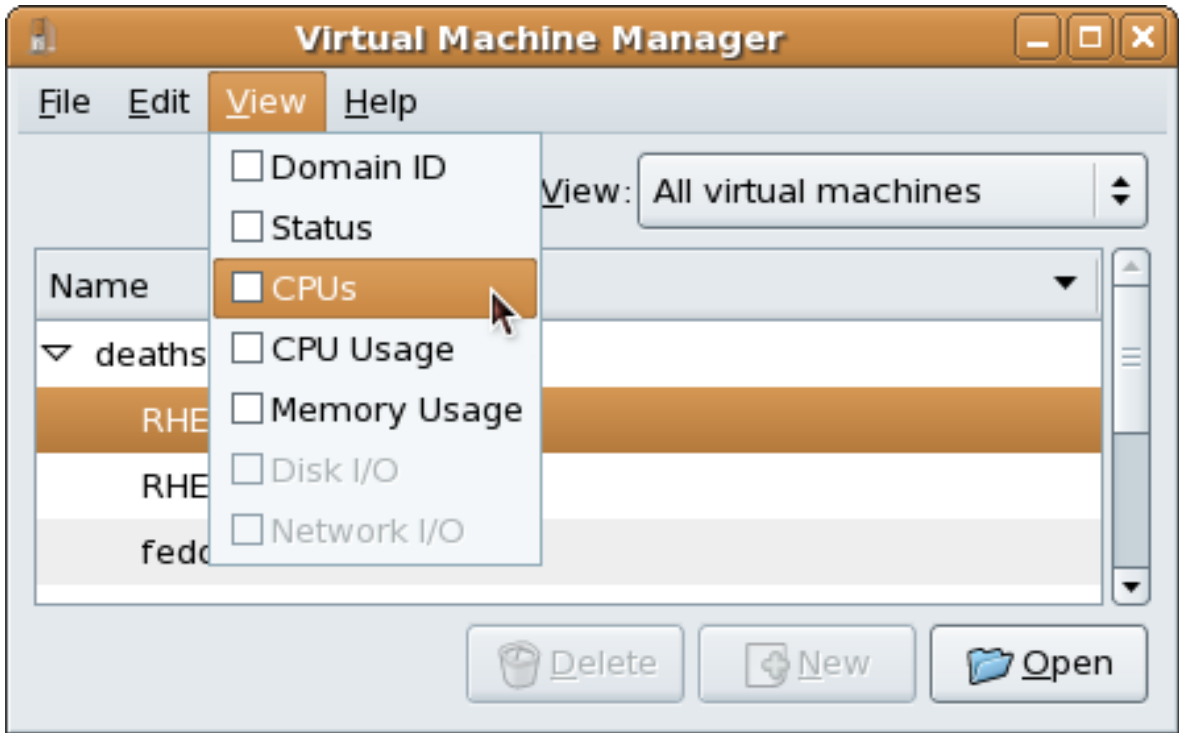


图16.22. 选择虚拟 CPU 选项

- 虚拟机管理器将列出系统中所有虚拟机的虚拟 CPU。

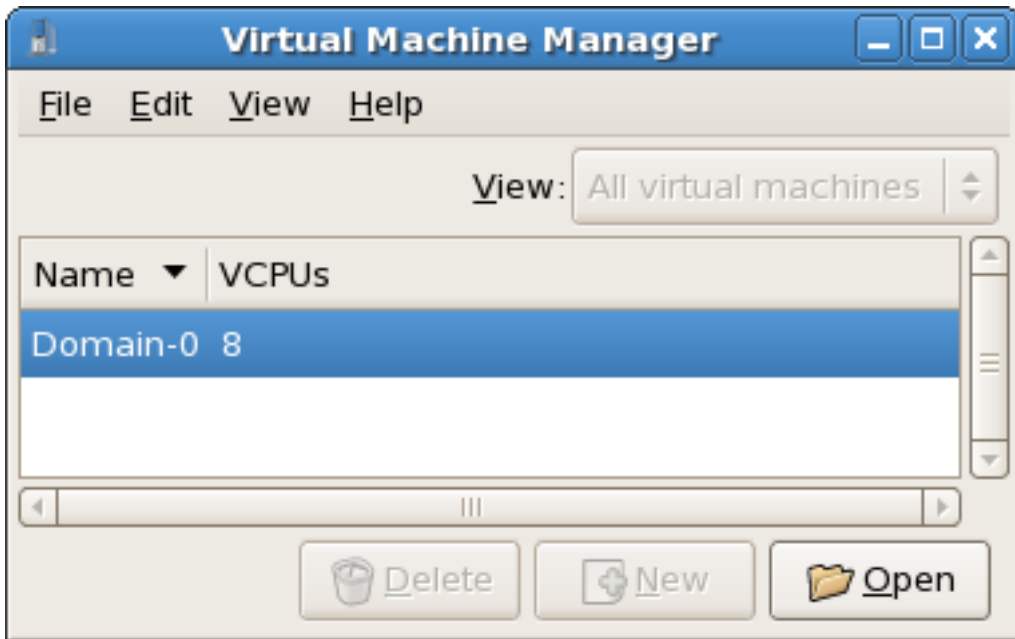


图16.23. 显示虚拟 CPU

## 16.12. 显示 CPU 用量

查看系统中所有虚拟机的 CPU 使用情况：

- 在「查看」菜单中选择「CPU 用量」复选框。

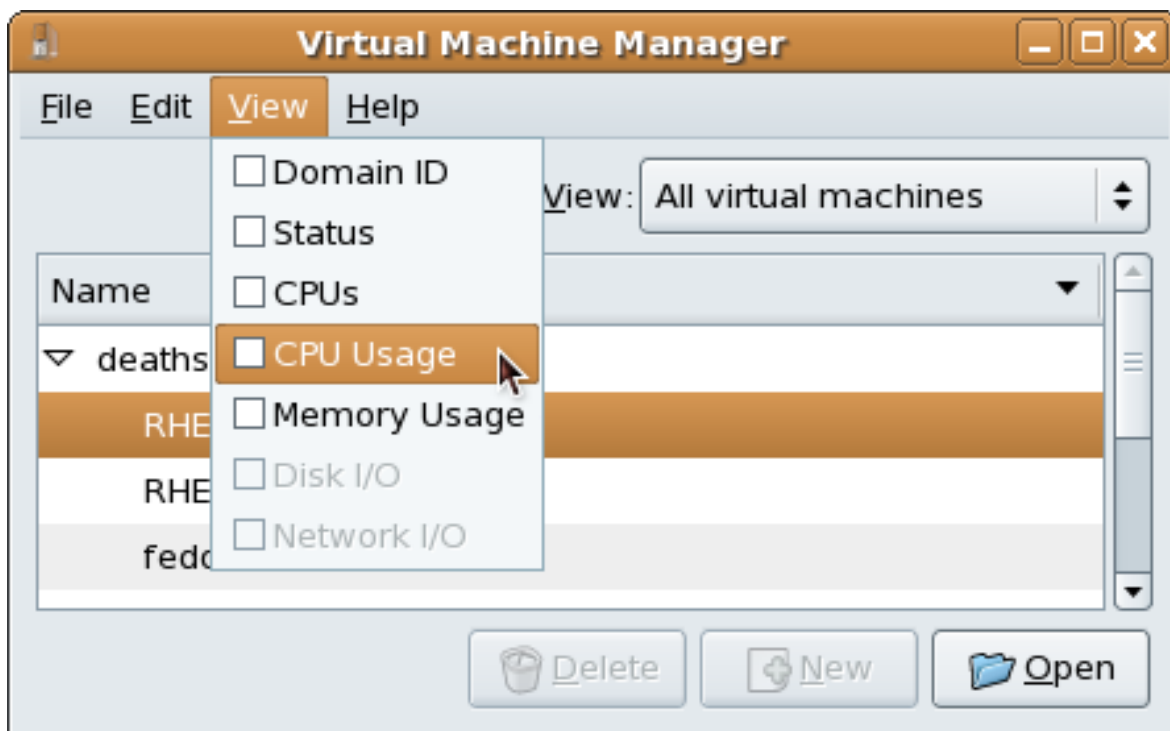


图 16.24. 选择 CPU 用量

- 虚拟机管理器将列出系统中所有虚拟机的 CPU 使用百分比。

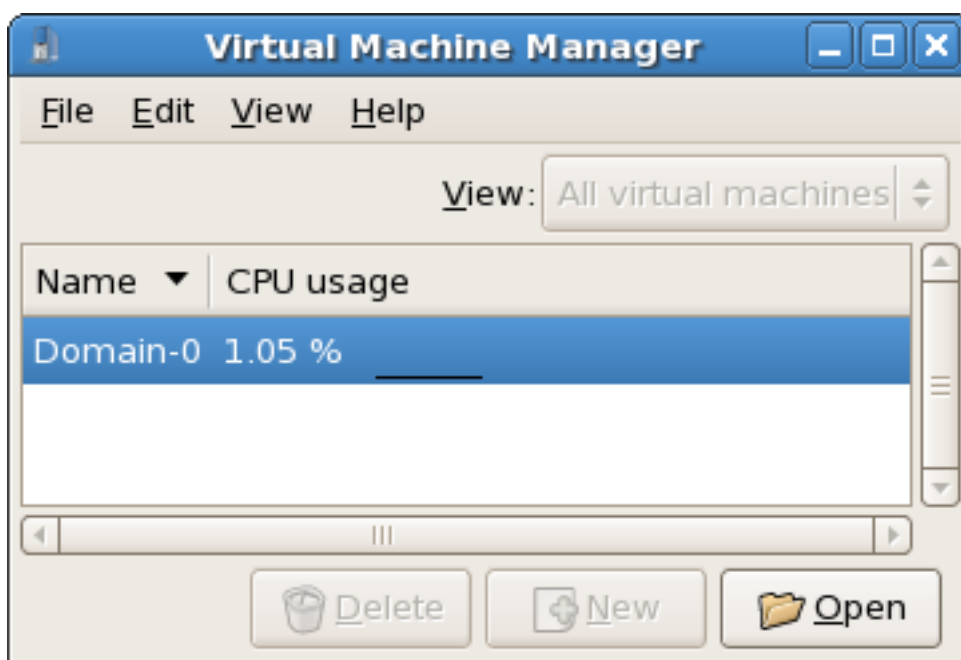


图 16.25. 显示 CPU 用量

## 16.13. 显示内存用量

查看系统中所有虚拟机的内存用量：

- 在「查看」菜单中选择「内存用量」复选框。

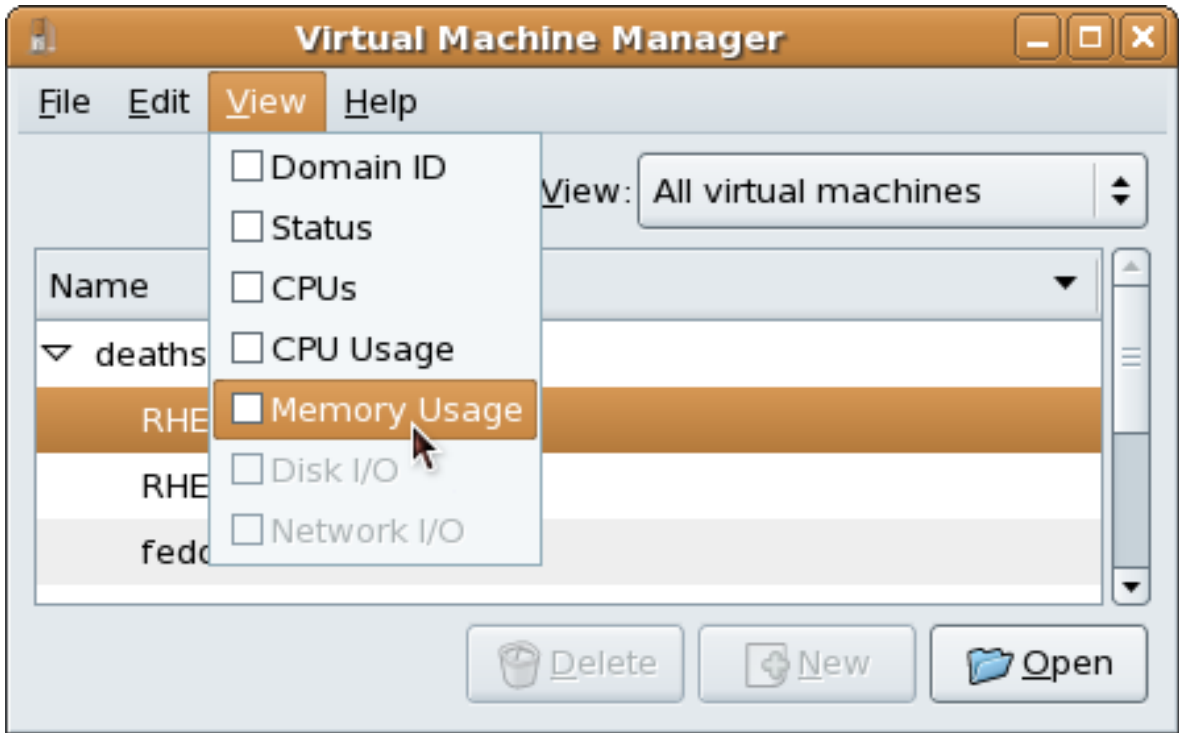


图16.26.0显示内存用量

- 虚拟机管理器将列出系统中所有虚拟机的内存（以 MB 为单位）使用百分比。

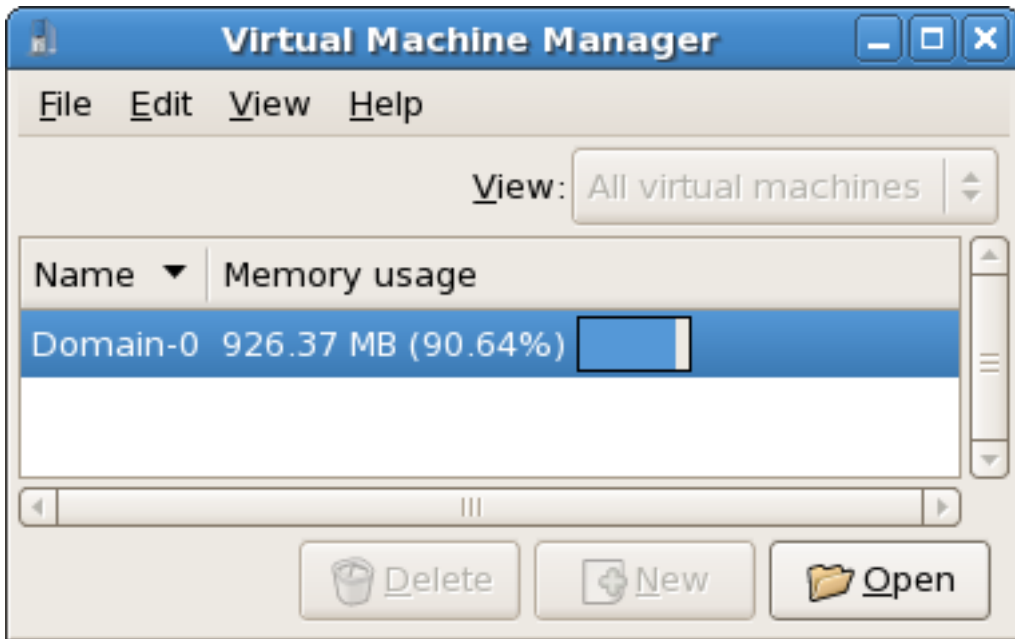


图16.27.0显示内存用量

## 16.14.0 管理虚拟网络

在您的系统中配置虚拟网络：

- 在「编辑」菜单中选择「主机详情」。



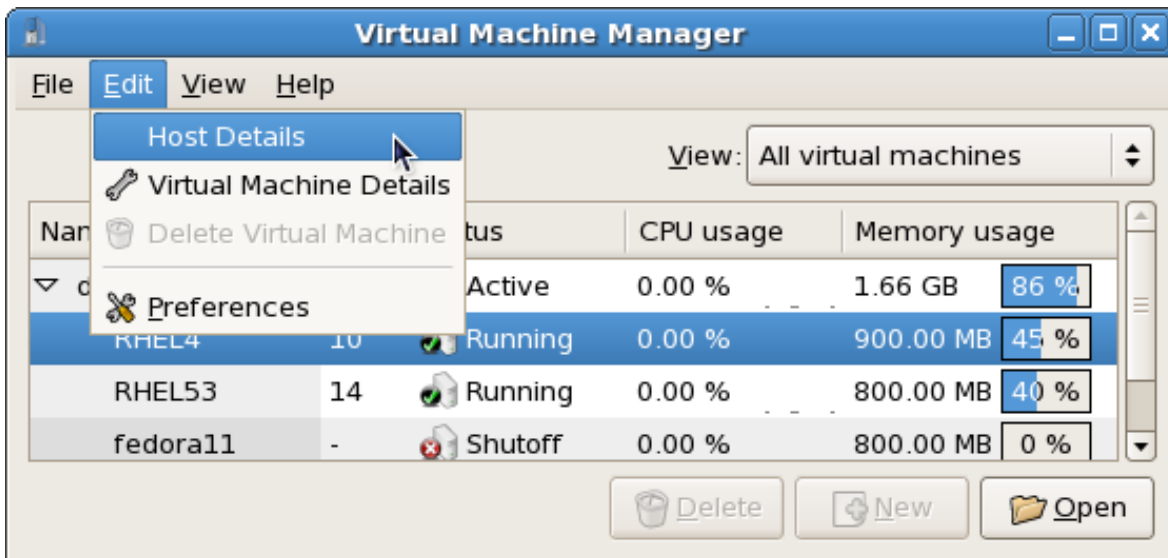


图 16.28.0 选择主机详情

2. 这将打开「主机详情」菜单。点击「虚拟网络」标签。

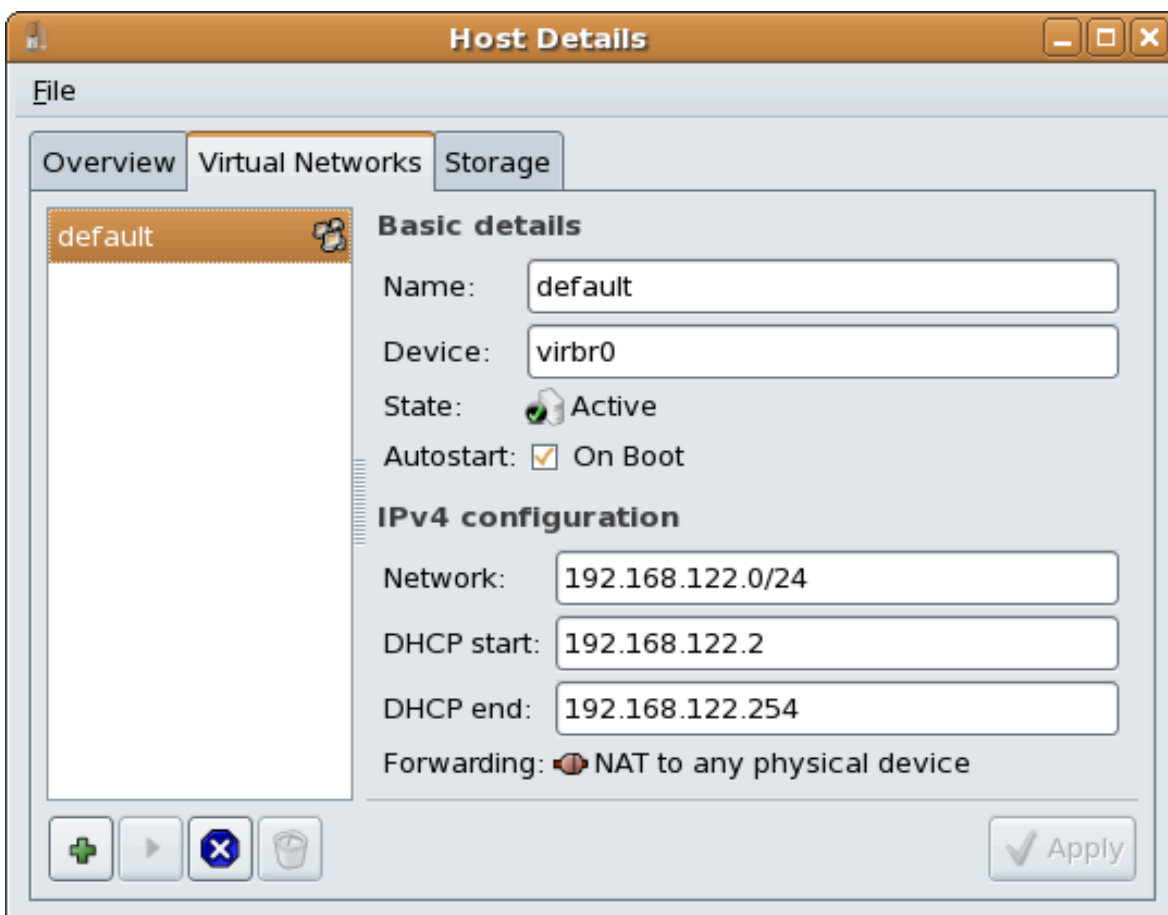


图 16.29.0 虚拟网络配置

3. 所有可用的虚拟网络都列在左首的菜单条里。您可以选择其中一个虚拟网络并编辑其配置。

## 16.15. 创建虚拟网络

在您的系统中创建新的虚拟网络：

1. 打开「主机详情」菜单（参考 16.14 “”）并点击 添加 按钮。

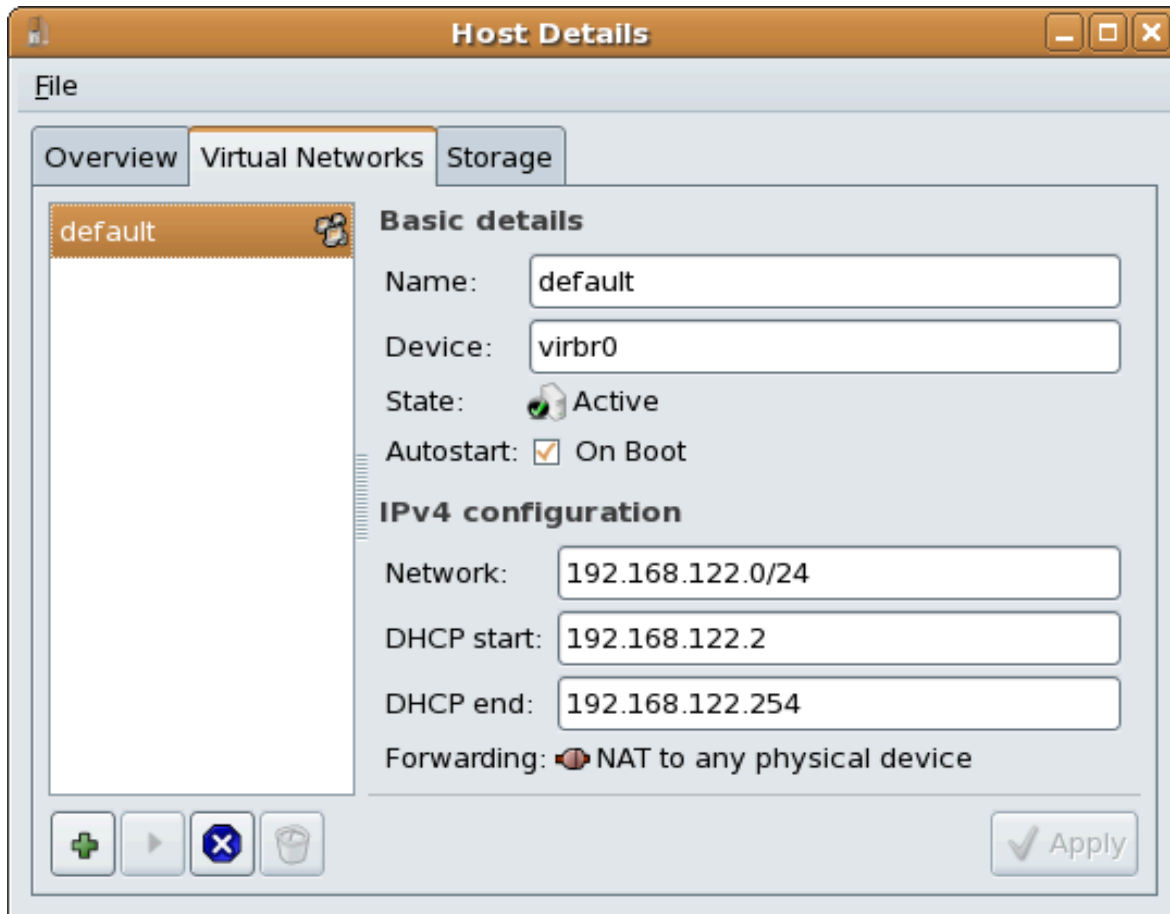


图16.30. 虚拟网络配置

这将打开「创建新的虚拟网络」菜单。点击 前进 按钮继续。

## Creating a new virtual network

This assistant will guide you through creating a new virtual network. You will be asked for some information about the virtual network you'd like to create, such as:

- A **name** for your new virtual network
- The IPv4 **address** and **netmask** to assign
- The **address range** from which the **DHCP** server will allocate addresses for virtual machines
- Whether to **forward** traffic to the physical network

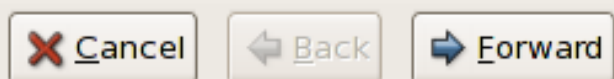


图 16.31. 创建新的虚拟网络

2. 输入新虚拟系统的名字并点击 前进。



图16.32. 命名虚拟网络

3. 为您的虚拟系统输入 IPv4 地址空间并点击 前进。

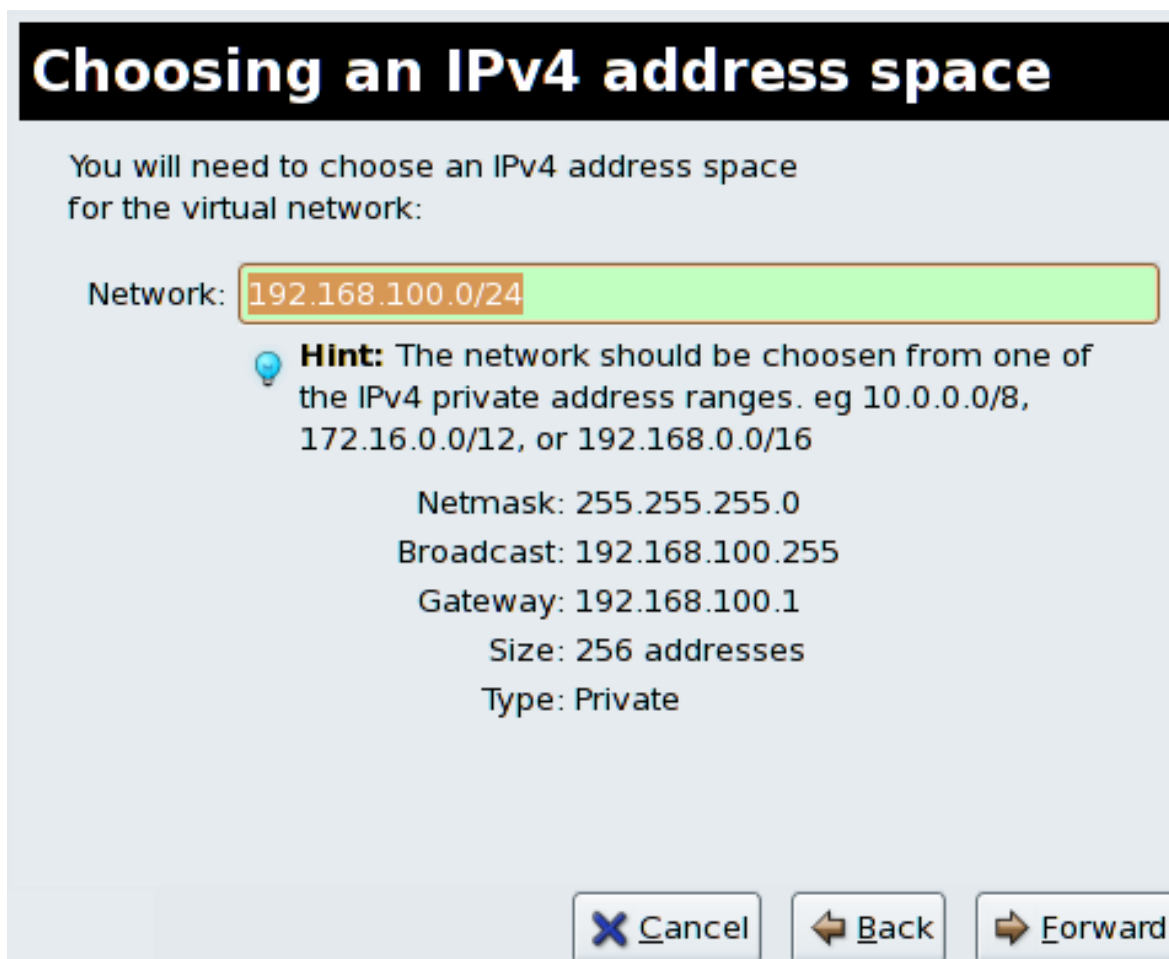


图 16.33.0 选择 IPv4 地址空间

4. 通过指定 IP 地址的「起始」和「终止」范围为您的虚拟网络定义 DHCP 范围。点击 前进 继续。

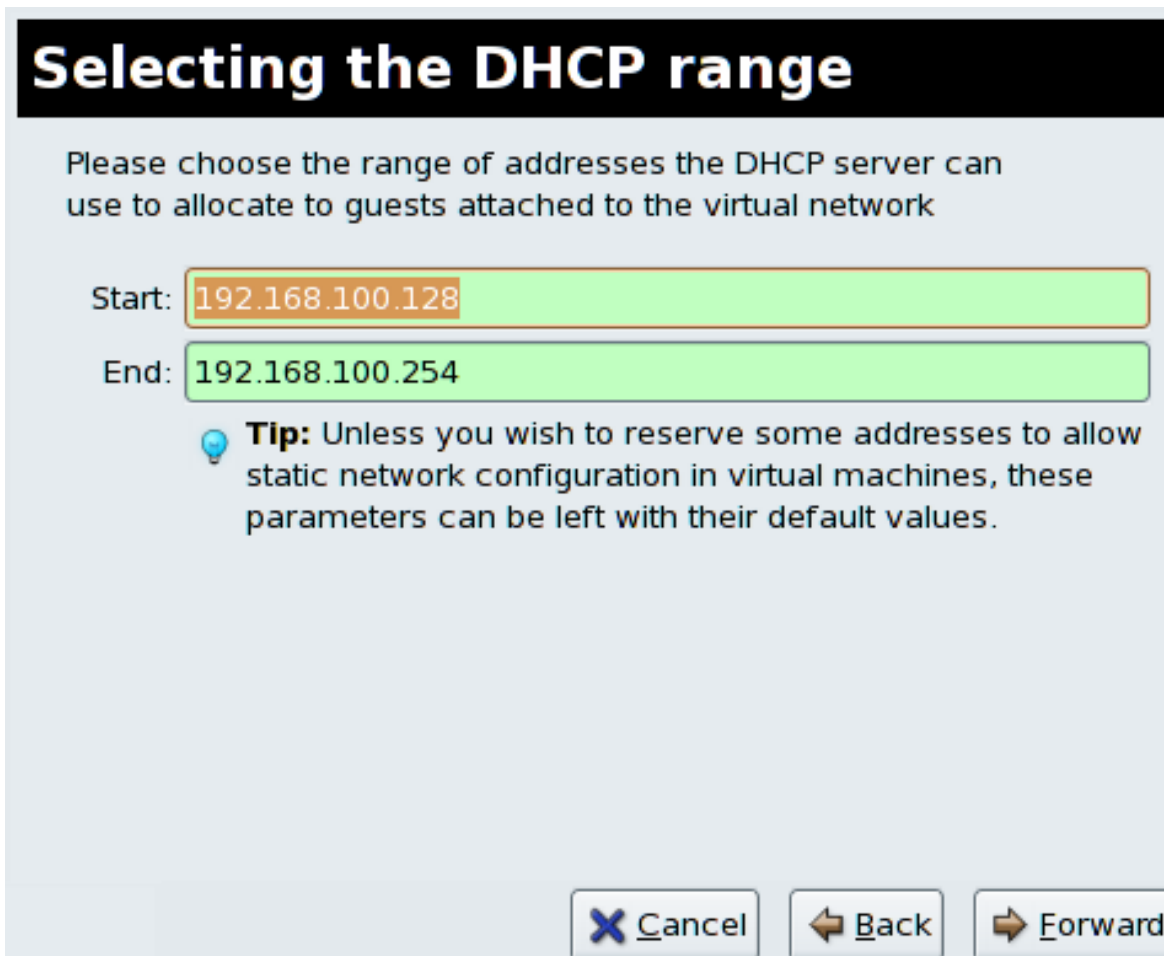


图16.34. 选择DHCP 范围

5. 选择虚拟网络应该怎样连接至物理网络。

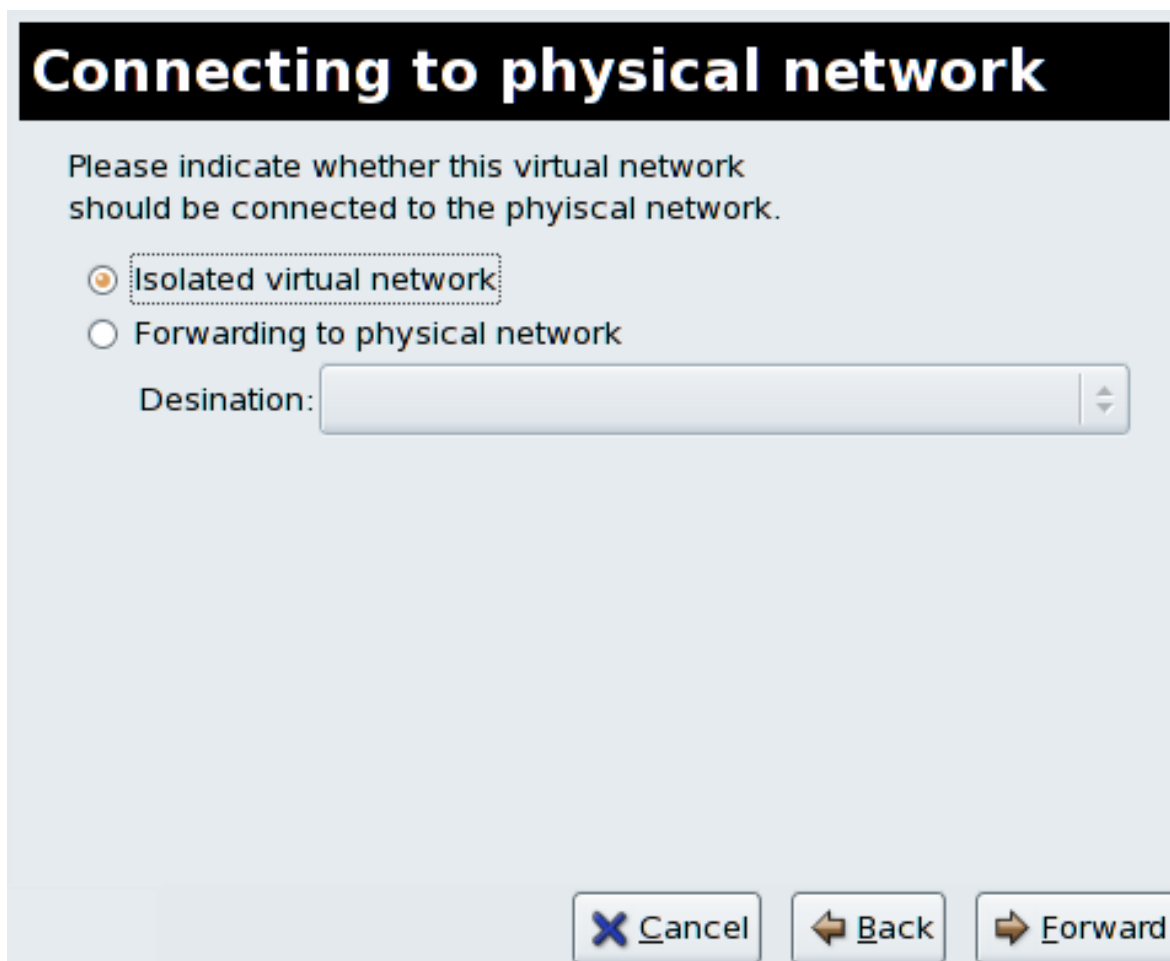


图 16.35. 连接至物理网络

如果您选择了「转发至物理网络」，请选择「目的网络」是否应该「NAT 到任意物理设备」或「NAT 到物理设备 eth0」。

Click Forward to continue.

6. 现在您可以创建网络了。检查您的网络配置并点击 完成。

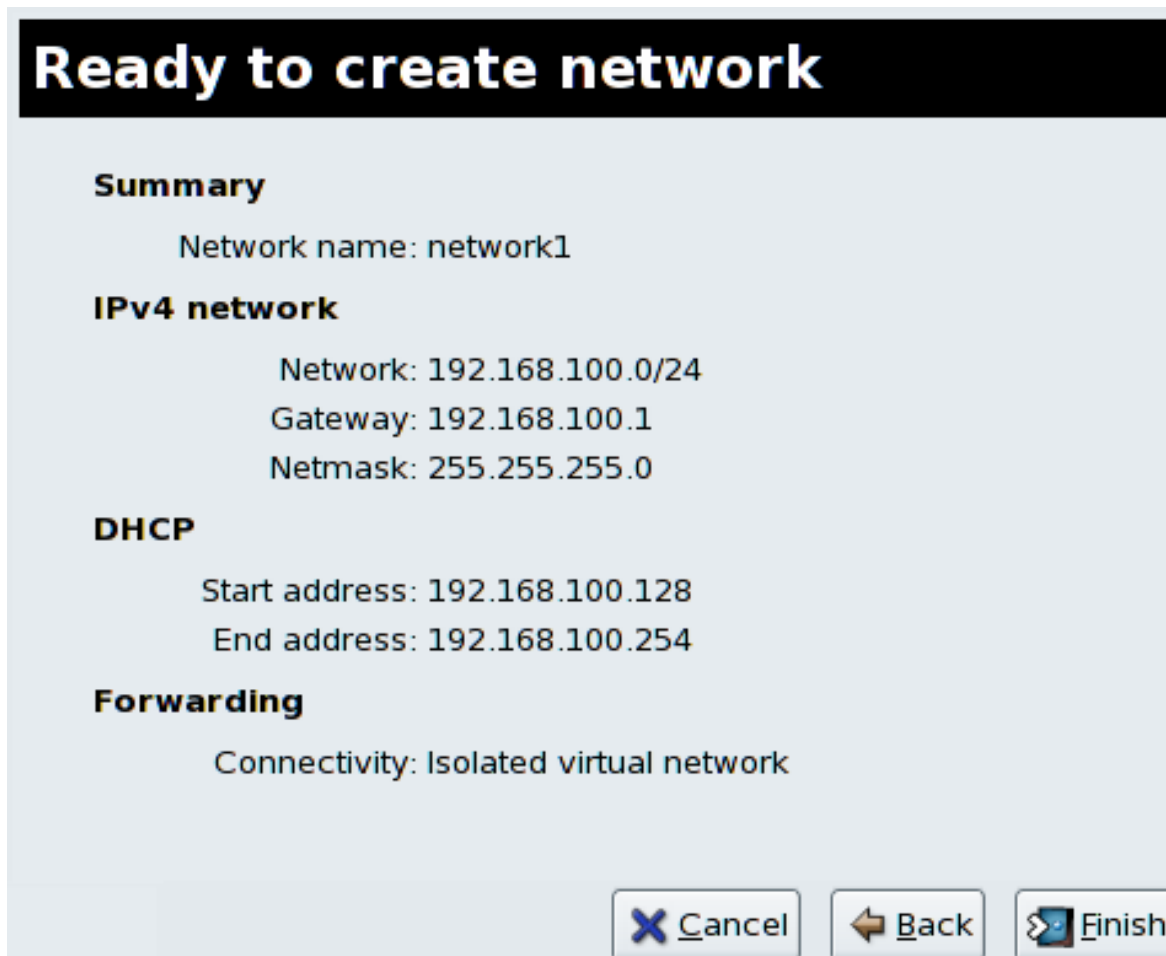


图16.36. 准备创建网络

7. 现在这个新的虚拟网络出现在「主机详情」菜单中的「虚拟网络」标签页。



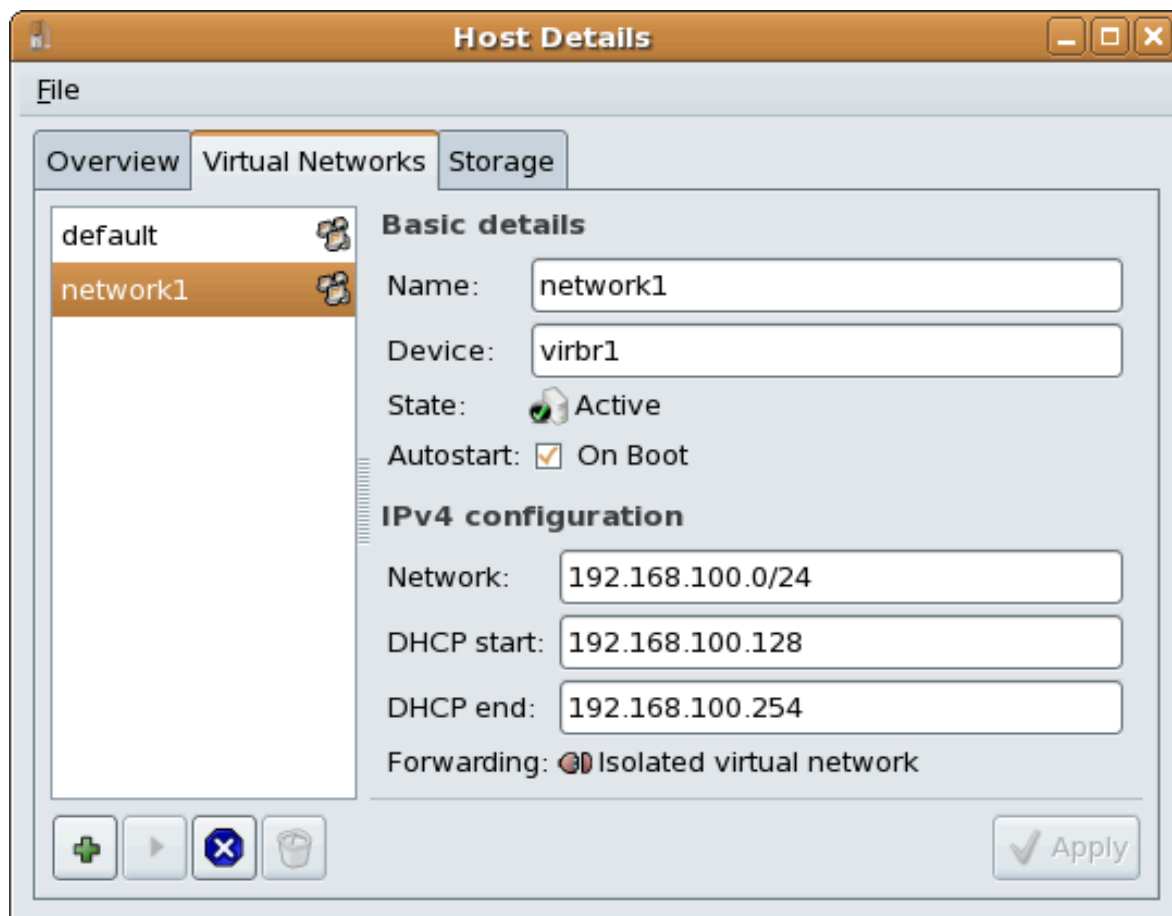


图 16.37.0 新的虚拟网络现在可用了

---

---

## 部分 V. Tips and Tricks

### 提高生产率的提示和窍门

这些章节包含改进虚拟化性能、范围及稳定性的有用窍门和提示。

---

---

## 提示及陷阱

这些章节包含用来改进虚拟化性能、范围和稳定性的有用提示和窍门。

### 17.1.0 自动启动客户端

这部分论述了如何在主机系统引导阶段自动启动虚拟客户端。

这个手里使用 `virsh` 设置客户端，使用 `TestServer` 在主机引导时自动启动。

```
# virsh autostart TestServer
Domain TestServer marked as autostarted
```

现在使用该主机自动启动该客户端。

请使用 `--disable` 停止客户端自动引导

```
# virsh autostart --disable TestServer
Domain TestServer unmarked as autostarted
```

不再使用该主机自动启动该客户端。

### 17.2.0 在 KVM 和 Xen 管理程序间互换

这部分论述了在 KVM 和 Xen 管理程序间互换。

Fedora 每次只支持一个活跃管理程序。



#### 在管理程序间迁移虚拟化客户端

目前，没有程序可进行从基于 Xen 客户端向 KVM 的切换或者从基于 KVM 客户端向 Xen 的切换。客户端只能使用创建它们的管理程序类型。

#### 17.2.1.0 Xen 到 KVM

下面的步骤论述的是从 Xen 管理程序改为 KVM 管理程序。这个步骤假设安装并启用了 `kernel-xen` 软件包。

1. 安装 KVM 软件包  
如果您还没有安装 `kvm` 软件包，则请安装它。

```
# yum install kvm
```

2. 确定所用内核  
可能安装了 `kernel-xen` 软件包。请使用 `uname` 命令确定正在运行的内核：

```
$ uname -r
2.6.23.14-107.fc8xen
```

目前在系统中运行的是 "2.6.23.14-107.fc8xen" 内核。如果运行的是默认 "2.6.23.14-107.fc8xen"，请跳过该分步。

- 将 Xen 内核改为默认内核

grub.conf 文件确定引导的是哪个内核。要更改默认内核请编辑如下所示的 /boot/grub/grub.conf 文件。

```
default=1
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora (2.6.23.14-107.fc8)
    root (hd0,0)
    kernel /vmlinuz-2.6.23.14-107.fc8 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
    initrd /initrd-2.6.23.14-107.fc8.img
title Fedora (2.6.23.14-107.fc8xen)
    root (hd0,0)
    kernel /xen.gz-2.6.23.14-107.fc8
    module /vmlinuz-2.6.23.14-107.fc8xen ro root=/dev/VolGroup00/LogVol00 rhgb quiet
    module /initrd-2.6.23.14-107.fc8xen.img
```

注意 default=1 参数。提示 GRUB 引导装载程序引导第二个条目，即 Xen 内核。将默认值改为 0（或者代表默认内核的数字）：

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora (2.6.23.14-107.fc8)
    root (hd0,0)
    kernel /vmlinuz-2.6.23.14-107.fc8 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
    initrd /initrd-2.6.23.14-107.fc8.img
title Fedora (2.6.23.14-107.fc8xen)
    root (hd0,0)
    kernel /xen.gz-2.6.23.14-107.fc8
    module /vmlinuz-2.6.23.14-107.fc8xen ro root=/dev/VolGroup00/LogVol00 rhgb quiet
    module /initrd-2.6.23.14-107.fc8xen.img
```

### 3. 重启载入新内核

重启该系统。计算机将使用默认内核重启，使用该模块可自动载入 KVM 模块。验证 KVM 请运行：

```
$ lsmod | grep kvm
kvm_intel      85992 1
kvm           222368 2 kvm,kvm_intel
```

如果一切正常，则会出现 kvm 模块和 kvm\_intel 模块或者 kvm\_amd 模块之一。

## 17.2.2. KVM 到 Xen

以下步骤论述从 KVM 管理程序改为 Xen 管理程序。这个过程假设安装并启用了 *kvm* 软件包。

### 1. 安装 Xen 软件包

如果您还没有，请安装 *kernel-xen* 和 *xen* 软件包。

```
# yum install kernel-xen xen
```

可能安装了 *kernel-xen* 软件包但被禁用。

### 2. 确定所用内核

使用 `uname` 命令确定正在运行的内核。

```
$ uname -r
2.6.23.14-107.fc8
```

出现的内核 "2.6.23.14-107.fc8xen" 就是正在系统中运行的内核。这是默认内核。如果内核在结尾处有 `xen`（例如 2.6.23.14-107.fc8xen），那么是 Xen 内核正在运行，这样您就可以跳过下面的子步骤。

- 将默认内核改为 Xen 内核

`grub.conf` 文件确定引导的是哪个内核。要更改默认内核请编辑如下所示的 `/boot/grub/grub.conf` 文件。

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora (2.6.23.14-107.fc8)
    root (hd0,0)
    kernel /vmlinuz-2.6.23.14-107.fc8 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
    initrd /initrd-2.6.23.14-107.fc8.img
title Fedora (2.6.23.14-107.fc8xen)
    root (hd0,0)
    kernel /xen.gz-2.6.23.14-107.fc8
    module /vmlinuz-2.6.23.14-107.fc8xen ro root=/dev/VolGroup00/LogVol00 rhgb quiet
    module /initrd-2.6.23.14-107.fc8xen.img
```

请注意 `default=0` 参数。这个参数让 GRUB 引导装载程序引导第一个条目，即默认内核。将默认改为 **1**（或者其它代表 Xen 内核的数字）：

```
default=1
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora (2.6.23.14-107.fc8)
    root (hd0,0)
    kernel /vmlinuz-2.6.23.14-107.fc8 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
    initrd /initrd-2.6.23.14-107.fc8
```

```
title Fedora (2.6.23.14-107.fc8xen)
root (hd0,0)
kernel /xen.gz-2.6.23.14-107.fc8
module /vmlinuz-2.6.23.14-107.fc8xen ro root=/dev/VolGroup00/LogVol00 rhgb quiet
module /initrd-2.6.23.14-107.fc8xen.img
```

### 3. 重启载入新内核

重启系统。计算机将使用 Xen 内核重新启动。使用 `uname` 命令验证：

```
$ uname -r
2.6.23.14-107.fc8xen
```

如果在输出的结尾处有 `xen`，那么运行的就是 Xen 内核。

## 17.3. 使用 `qemu-img`

`qemu-img` 命令行工具是 Xen 和 KVM 用来格式化各种文件系统的。可使用 `qemu-img` 格式化虚拟客户端映像、附加存储设备以及网络存储。`qemu-img` 选项及用法如下。

### 格式化并创建新映像或者设备

创建新磁盘映像文件名为 `size`，格式为 `format`。

```
# qemu-img create [-6] [-e] [-b base_image] [-f format] filename [size]
```

如果指定了 `base_image`，则该映像只会记录与 `base_image` 的不同。在此无须指定任何大小。从来不需要修改 `base_image` 除非您使用 `commit` `monitor` 命令。

### 将现有映像转换成另一种格式

转换选项是将可识别格式转换为另一个映像格式。

命令格式：

```
# qemu-img convert [-c] [-e] [-f format] filename [-O output_format] output_filename
```

使用格式 `output_format` 将磁盘映像 `filename` 转换为磁盘映像 `output_filename`。可使用 `-e` 选项对该磁盘映像进行加密或者使用 `-c` 选项对其进行压缩。

只有格式 `qcow` 支持加密或者压缩。加密是只读的，就是说如果重新写入压缩的字段，那么会以未压缩数据形式重新写入。

加密法是使用非常安全的 128 位密钥的 AES 格式。使用长密码（16 个字符以上）获得最大程度的保护。

当使用可增大的映像格式，比如 `qcow` 或 `cow` 时，映像转换可帮助您获得较小的映像。在目的映像中可检测并压缩空白字段。

### 获得映像信息

`info` 参数显示磁盘映像信息。`info` 选项的格式如下：

```
# qemu-img info [-f format] filename
```



给出磁盘映像文件名信息。使用它可获得在磁盘中保留空间大小，它可能与显示的大小有所不同。如果在磁盘映像中保存有 vm 快照，则此时也会显示。

## 支持格式

映像格式通常是自动获取的。支持以下格式：

### raw

Raw 磁盘映像格式（默认）。这个格式的优点是可以简单、容易地导出到其它模拟器中。如果您的文件系统支持中断（例如在 Linux 中的 ext2 或者 ext3 以及 Windows 中的 NTFS），那么只有写入的字段会占用空间。使用 `qemu-img info` 了解 Unix/Linux 中映像或者 `ls -ls` 使用的实际大小。

### qcow2

QEMU 映像格式，最万能的格式。使用它可获得较小映像（如果您的系统不支持中断，例如在 Windows 中，它会很有用）、额外的 AES 加密法、zlib 压缩以及对多 VM 快照的支持。

### qcow

旧的 QEMU 映像格式。只用于与旧版本兼容。

### cow

写入映像格式的用户模式 Linux 副本。包含 cow 格式的目的只是为了与前面的版本兼容。它无法在 Windows 中使用。

### vmrk

VMware 3 和 4 兼容映像格式。

### cloop

Linux 压缩回送映像，只有在重复使用直接压缩的 CD-ROM 映像时有用，比如在 Knoppix CD-ROM 中。

## 17.4.0 KVM 中的过量使用

KVM 管理程序支持 CPU 和内存的过量使用。过量使用是分配超过系统中物理资源的虚拟 CPU 或者内存。如果过量使用 CPU，那么未利用的虚拟服务器或者桌面就可运行一些服务器，继而可以减少费用。



### Xen 支持

Xen 管理程序不支持 CPU 过量使用。在 Xen 管理程序中过量使用 CPU 可导致系统不稳定并使主机和虚拟客户端崩溃。

### 过量使用内存

大多数操作系统和应用程序不会总是使用 100% 可用 RAM。KVM 可利用这种行为让虚拟客户端使用超过物理可用的内存量。

在 KVM 中，虚拟机是 Linux 进程。KVM 管理程序中的客户端没有为其分配的物理 RAM 块，而是作为进程工作。每个进程在其需要时为其分配更多内存。KVM 使用这个方法在客户端操作系统需要更多或者较少内存时为其进行分配。客户端使用的物理内存只比虚拟操作系统要使用的物理内存稍多一些。

当物理内存消耗殆尽或者某个进程有时不活跃时，Linux 会将进程内存运动到 swap 中。Swap 通常是在硬盘或者固定状态驱动器中 Linux 用来扩展虚拟内存的分区。Swap 比 RAM 的速度稍慢。

因为 KVM 虚拟机是 Linux 进程，所以如果虚拟客户端处于空闲状态或者没有大量使用，则会将该客户端使用的内存放入 swap 中。内存使用量可超过 swap 和物理 RAM 的总和。如果虚拟客户端使用其全部 RAM

则会造成一些问题。如果虚拟机没有足够 swap 空间，则会启动将进程切换到 swap 中的 pdflush 进程，即清除进程。pdflush 会杀死进程来释放内存以便系统不至于崩溃。pdflush 可能会销毁导致文件系统出错虚拟客户端或者其他系统进程，并导致虚拟客户端无法引导。



### Warning

如果没有足够的可用 swap，客户端操作系统将强制关闭。这可能导致对客户端无法操作。要避免这种情况出现，请不要使用超过可用 swap 内存的量。

swap 分区是用来将未使用的内存交换到硬盘中以便提高内存性能。默认 swap 分区大小是根据 RAM 大小和过量使用比例来计算的。如果您要在 KVM 中过量使用内存，则建议您为 swap 分区分配较大的空间。推荐的过量使用比例为 50% (0.5)。计算方程式为：

$$(0.5 * \text{RAM}) + (\text{过量使用比例} * \text{RAM}) = \text{推荐 swap 大小}$$

红帽<sup>1</sup>有一篇论述了安全有效地确定 swap 分区大小的文章。

内存过量使用比例最多可以是系统中物理 RAM 量的十倍。这只能在某些应用程序负载中使用（例如：桌面虚拟化用量在 100% 以下）。设定过量使用比例不是一个固定的方程式，您必须为您的环境进行测试并优化该比例。

### 过量使用的虚拟 CPU

KVM 管理程序支持过量使用的虚拟 CPU。只要虚拟客户端负载限制允许即可过量使用虚拟 CPU。当过量使用 VCPU 时要小心，因为负载接近 100% 时可能导致释放请求或者不可用响应次数。

过量使用虚拟 CPU 的最佳时机是当每个虚拟客户端只有一个 VCPU 的时候。Linux 调度程序对这种类型的负载非常有效。KVM 在比例为 5 VPCU，负载不超过 100% 时应该可安全支持客户端。过量使用单一 VCPU 虚拟客户端不会有问题。

您过量使用的对称多进程客户端数量不能超过处理核的物理数量。例如：有四个 VCPU 的客户端不能在运行双核处理器的主机中运行。过量使用对称多进程客户端的数量如果超过处理核的物理数量将导致性能严重下降。

分配客户端 VCPU 的数量不超过物理核的数量是可以的，也可以正常工作。例如：在四核主机中运行有四个 VCPU 的虚拟客户端。未达到 100% 负载的客户端在这种设置中可有效工作。



### 总是先测试

在没有进行大规模测试前不要在产品环境中过量使用内存或者 CPU。使用 100% 内存或者进程资源的应用程序可能在过量使用的环境中变得不稳定。部署前必须进行测试。

## 17.5. 修改 /etc/grub.conf

这部分论述了如何安全且正确地修改您的 /etc/grub.conf 文件使其使用虚拟化内核。您必须使用 xen 内核应用 Xen 监控程序。复制您现有的 xen 内核条目，确定您复制了所有重要的行，否则您的系统将在引导时 panic (initrd 的长度将为 '0')。如果您需要 xen 监控程序特定值，您必须将其附加到您 grub 条目的 xen 行中。

<sup>1</sup> <http://kbase.redhat.com/faq/docs/DOC-15252>

以下输出结果是运行 *kernel-xen* 软件包的系统的 `grub.conf` 条目示例。在您的系统中的 `grub.conf` 文件可能会有所不同。示例中重要的是从 `title` 行到下一个新行之间的部分。

```
#boot=/dev/sda
default=0
timeout=15
#splashimage=(hd0,0)/grub/splash.xpm.gz hiddenmenu
serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
terminal --timeout=10 serial console

title Fedora (2,6,23,14-107.fc8xen)
    root (hd0,0)
    kernel /xen.gz-2,6,23,14-107.fc8 com1=115200,8n1
    module /vmlinuz-2,6,23,14-107.fc8xen ro root=/dev/VolGroup00/LogVol00
    module /initrd-2,6,23,14-107.fc8xen.img
```



### 有关编辑 `grub.conf` 中重要的一点.....

如果您的 `grub.conf` 在从示例中复制前已经进行过手动编辑，那么它看起来可能完全不同。

要在引导时为您的主机系统分配 256MB 内存，您需要在您 `grub.conf` 文件的 `xen` 行中附加 `dom0_mem=256M`。前面示例的 `grub` 配置文件的修订版本如下：

```
#boot=/dev/sda
default=0
timeout=15
#splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
terminal --timeout=10 serial console

title Fedora (2,6,23,14-107.fc8xen)
    root (hd0,0)
    kernel /xen.gz-2,6,23,14-107.fc8 com1=115200,8n1 dom0_mem=256MB
    module /vmlinuz-2,6,23,14-107.fc8xen ro
    root=/dev/VolGroup00/LogVol00
    module /initrd-2,6,23,14-107.fc8xen.img
```

## 17.6.0 验证虚拟化扩展

使用这部分确定您的系统是否有硬件虚拟化扩展。虚拟化扩展（Intel VT 或者 AMD-V）是全虚拟化所需要的。



### 我可以不使用没有虚拟化扩展的虚拟化吗？

如果没有出现硬件虚拟化扩展，您可以使用 *fedora kernel-xen* 软件包附带的 Xen 半虚拟化。

运行以下命令确定 CPU 虚拟化扩展可用：

```
$ grep -E 'svm|vmx' /proc/cpuinfo
```

下面的输出结果包含 `vmx` 条目，表明是带 Intel VT 扩展的 Intel 处理器：

```
flags : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush
      dts acpi mmx fxsr sse sse2 ss ht tm syscall lm constant_tsc pni monitor ds_cpl
      vmx est tm2 cx16 xtpr lahf_lm
```

下面的输出结果包含 `svm` 条目，表明是带 AMD-V 扩展的 AMD 处理器：

```
flags : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush
      mmx fxsr sse sse2 ht syscall nx mmxext fxsr_opt lm 3dnowext 3dnow pni cx16
      lahf_lm cmp_legacy svm cr8legacy ts fid vid ttp tm stc
```

系统中的每个超线程、核或者 CPU 可能出现多个 “flags:” 内容。

可能在 BIOS 中禁用了虚拟化扩展。如果没有出现扩展或者全虚拟化无法工作，请参考 [19.1](#)，“[BIOS 设置](#)”。

## 17.7. 鉴别客户端类型及使用

下面的脚本可以鉴别系统是半虚拟客户端、全虚拟客户端还是裸机主机。

```
#!/bin/bash
declare -i IS_HVM=0
declare -i IS_PARA=0
check_hvm()
{
  IS_X86HVM="$(strings /proc/acpi/dsdt | grep int-xen)"
  if [ x"${IS_X86HVM}" != x ]; then
    echo "Guest type is full-virt x86hvm"
    IS_HVM=1
  fi
}
check_para()
{
  if $(grep -q control_d /proc/xen/capabilities); then
    echo "Host is dom0"
    IS_PARA=1
  else
    echo "Guest is para-virt domU"
    IS_PARA=1
  fi
}
if [ -f /proc/acpi/dsdt ]; then
  check_hvm
fi
if [ ${IS_HVM} -eq 0 ]; then
```

```

if [ -f /proc/xen/capabilities ]; then
    check_para
fi
fi
if [ ${IS_HVM} -eq 0 -a ${IS_PARA} -eq 0 ]; then
    echo "Baremetal platform"
fi

```



### 检查主机

要检查主机请使用 `virsh capabilities` 命令。

## 17.8. 生成新的唯一 MAC 地址

在有些情况下您需要为客户端生成新的唯一 **MAC** 地址。完成本文时还没有可用来生成新 MAC 地址的命令行工具。下面提供的脚本可以为您的客户端生成新的 MAC 地址。将该脚本在您的客户端中保存为 `macgen.py`。现在您可以在该目录中使用 `./macgen.py` 运行该脚本生成新的 MAC 地址。输出应类似如下：

```

$ ./macgen.py
00:16:3e:20:b0:11

#!/usr/bin/python
# macgen.py script to generate a MAC address for virtualized guests on Xen
#
import random
#
def randomMAC():
    mac = [ 0x00, 0x16, 0x3e,
           random.randint(0x00, 0x7f),
           random.randint(0x00, 0xff),
           random.randint(0x00, 0xff) ]
    return ':'.join(map(lambda x: "%02x" % x, mac))
#
print randomMAC()

```

### 为您的客户端生成新 MAC 地址的另一种方法

您还可以使用 `python-virtinst` 的内嵌模块生成用于客户端配置文件的新 MAC 地址以及 UUID：

```

# echo 'import virtinst.util; print\
virtinst.util.uuidToString(virtinst.util.randomUUID())' | python
# echo 'import virtinst.util; print virtinst.util.randomMAC()' | python

```

以上脚本还可以作为脚本文件使用，如下。

```

#!/usr/bin/env python
# -*- mode: python; -*-
print ""

```

```
print "New UUID:"
import virtinst.util; print virtinst.util.uuidToString(virtinst.util.randomUUID())
print "New MAC:"
import virtinst.util; print virtinst.util.randomMAC()
print ""
```

## 17.9. 非常安全的ftpd

vsftpd 为半虚拟客户端或者其它数据库提供对安装树的访问。如果您在安装服务器的过程中还没有安装 vsftpd，您可以从您安装介质的 Server 目录中获得 RPM 软件包，并使用 `rpm -ivh vsftpd*.rpm` 安装（请注意：RPM 软件包必须位于您的当前目录中）。

1. 要配置 vsftpd，请使用 `vi` 编辑 `/etc/passwd`，并将 ftp 用户主目录改为您要为您的半虚拟客户端保存安装树的目录。FTP 用户条目示例如下：

```
ftp:x:14:50:FTP User:/xen/pub:/sbin/nologin
```

2. 要使 vsftpd 在系统引导时自动启动，请使用 `chkconfig` 程序启用 vsftpd 的自动启动。
3. 使用 `chkconfig --list vsftpd` 确定没有启用 vsftpd：

```
$ chkconfig --list vsftpd
vsftpd    0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

4. 运行 `chkconfig --levels 345 vsftpd on` 为运行级别 3、4 和 5 自动启用 vsftpd。
5. 使用 `chkconfig --list vsftpd` 命令确定在系统引导时会自动启用 vsftpd：

```
$ chkconfig --list vsftpd
vsftpd    0:off 1:off 2:off 3:on 4:on 5:on 6:off
```

6. 使用 `service vsftpd start vsftpd` 启动 vsftpd 服务：

```
$service vsftpd start vsftpd
Starting vsftpd for vsftpd:      [ OK ]
```

## 17.10. 配置 LUN 持久性

这部分论述了如何在客户端以及主机（可以是多路径，也可以不是）部署 LUN 持久性。

### 在非多路径系统中部署 LUN 持久性

如果您的系统没有使用多路径，您可以使用 `udev` 部署 LUN 持久性。在您的系统中部署 LUN 持久性前，请确定您有正确的 UUID。您得到它们后，即可编辑位于 `/etc` 目录的 `scsi_id` 文件来配置 LUN 持久性。当您在文本编辑器中打开此文件时，您必须注释出这一行：

```
# options=-b
```

然后用这个参数来代替它：

```
# options=-g
```

这样可以 `udev` 监控所有系统 SCSI 设备以返回 UUID。要决定该系统的 UUID，请使用 `scsi_id` 命令：

```
# scsi_id -g -s /block/sdc
*3600a0b80001327510000015427b625e*
```

输出中的这个长字符串就是 UUID。在您向系统中添加新设备时，UUID 不会改变。请为每个设备获得一个 UUID 以便为设备创建规则。要创建新设备规则，请编辑位于 `/etc/udev/rules.d` 目录的 `20-names.rules` 文件。该设备命名规则的格式如下：

```
# KERNEL="sd*", BUS="scsi", PROGRAM="sbin/scsi_id", RESULT="UUID", NAME="devicename"
```

使用以上搜索出的 UUID 条目替换您现有 `UUID` 和 `devicename`。该规则应类似如下：

```
KERNEL="sd*", BUS="scsi", PROGRAM="sbin/scsi_id",
RESULT="3600a0b80001327510000015427b625e", NAME="mydevicename"
```

这可使系统启用所有与格式 `/dev/sd*` 匹配的设备来检查给定的 UUID。当找到第一个匹配的设备时，它会生成一个设备节点，名为 `/dev/devicename`。在这个示例中，该设备节点为 `/dev/mydevice`。最后，请在 `/etc/rc.local` 文件中添加这一行：

```
/sbin/start_udev
```

### 部署多路径 LUN 持久性

要在多路径环境中部署 LUN 持久性，您必须为多路径设备定义别名。在本示例中，您必须编辑位于 `/etc/` 目录中的 `multipath.conf` 文件来定义四个设备别名：

```
multipath {
    wwid    3600a0b80001327510000015427b625e
    alias   oramp1
}
multipath {
    wwid    3600a0b80001327510000015427b6
    alias   oramp2
}
multipath {
    wwid    3600a0b80001327510000015427b625e
    alias   oramp3
}
multipath {
    wwid    3600a0b80001327510000015427b625e
    alias   oramp4
}
```

这定义了四个 LUN：`/dev/mpath/oramp1`、`/dev/mpath/oramp2`、`/dev/mpath/oramp3` 和 `dev/mpath/oramp4`。这些设备将位于 `/dev/mpath` 目录。这些 LUN 名称会在重启后保留，因为在 LUN 的 `wwid` 中创建了别名。

## 17.11. 为客户端禁用 SMART 磁盘监控

可禁用 SMART 磁盘监控，因为我们是在虚拟磁盘中运行，且物理存储是由主机管理的：

```
/sbin/service smartd stop
/sbin/chkconfig --del smartd
```

## 17.12. 克隆客户端配置文件

您可以复制现有的配置文件来生成一个全新的客户端。您必须修改客户端配置文件的 `name` 参数。此时会在管理程序中出现一个新的唯一名称，且可由管理软件查看。您还必须生成一个全新的 UUID，可以使用 `uuidgen` 命令。然后在 `vif` 条目中您必须为每个客户端定义一个唯一的 MAC 地址（如果您是从现有客户端中复制客户端配置，您可以生成一个脚本来进行处理）。有关 `xenbr` 桥接信息，如果您将一个现有客户端配置文件移动到一个新的主机中，您必须更新 `xenbr` 条目来与您的本地联网配置匹配。对于设备条目，您必须修改 `'disk='` 部分中的条目使其指向正确的客户端映像。

您还必须在您的客户端中修改这些系统配置设定。您必须修改 `/etc/sysconfig/network` 文件中的 `HOSTNAME` 条目使其与新客户端的主机名匹配。

您必须修改 `/etc/sysconfig/network-scripts/ifcfg-eth0` 文件中的 `HWADDR` 地址使其与 `ifconfig eth0` 文件的输出匹配，同时如果您要使用静态 IP 地址，您还必须修改 `IPADDR` 条目。

## 17.13. 复制现有客户端及其配置文件

这部分论述了复制现有配置文件来创建一个新的客户端。您必须了解您客户端配置文件中的关键参数，并对其进行修改来成功复制客户端。

### name

管理程序必须了解您的客户端名称，并在管理工具中显示。这个条目应该是您的系统特有的。

### uuid

客户端的特有名称，使用 `uuidgen` 命令可重新生成一个新的 UUID。UUID 输出示例：

```
$ uuidgen
a984a14f-4191-4d14-868e-329906b211e5
```

### vif

- **MAC** 必须为每个客户端定义唯一的 MAC 地址，如果使用标准工具这个操作会自动完成。如果您要从现有客户端中复制客户端配置，您可以使用脚本 `17.8` 中的 `"MAC"`。
- 如果您要将现有客户端配置文件移动或者重复到新的主机中，您必须确定调整了 `xenbr` 条目以符合您的本地联网配置（您可以使用 `brctl show` 命令获得桥接信息）。
- 设备条目，请确定您调整了 `disk=` 部分的条目使其指向正确的客户端映像。

现在请在您的客户端中调整系统配置设置：

```
/etc/sysconfig/network
将 HOSTNAME 条目改为客户端的新 hostname。
```

```
/etc/sysconfig/network-scripts/ifcfg-eth0
• 用 ifconfig eth0 输出中的硬件地址替换 HWADDR 地址。
```



- 如果使用静态 IP 地址，请修改 IPADDR 条目。



## 生成自定义 libvirt 脚本

这部分将旨在为使用 libvirt 编写自定义脚本的程序员和系统管理员提供一些有用信息。

**17** 建议希望使用 libvirt 编写新应用程序的程序员阅读。

### 18.1. 使用带 virsh 的 XML 配置文件

virsh 可以处理 XML 配置文件。您可在使用特殊选项编写大量部署脚本时利用这个优点。您可以添加在 XML 文件中定义的设备来运行半虚拟客户端。例如，要添加一个 ISO 文件作为 hdc 运行客户端，会生成 XML 文件：

```
# cat satelliteiso.xml
<disk type="file" device="disk">
  <driver name="file"/>
  <source file="/var/lib/libvirt/images/rhn-satellite-5.0.1-11-redhat-linux-as-i386-4-embedded-
oracle.iso"/>
  <target dev="hdc"/>
  <readonly/>
</disk>
```

运行 virsh attach-device 命令将 ISO 文件作为 hdc 添加到名为 "satellite" 的客户端中：

```
# virsh attach-device satellite satelliteiso.xml
```



---

## 部分 VI. Troubleshooting

### 故障排除和问题解决介绍

下面的章节提供帮助您在虚拟化过程中可能遇到问题进行故障排除的信息。



#### 虚拟化问题的重要注释

由于不断的开发会生成并修复 bug，所以您的具体问题可能没有在此出现。大多数最新的已知 bug 列表以及问题和 bug 修复，请参考您使用版本和硬件构架的 Fedora。在线文档位于 Fedora 网站：<http://docs.fedoraproject.org>

---

---

# Troubleshooting

本章涵盖了 Fedora 虚拟化中的一般问题及解决方法。

## 19.1. 回送设备错误

如果使用基于文件的客户端映像，您可能必须要增加配置的回送设备数量。默认配置最多允许 8 个活跃的回送设备。如果需要 8 个以上的基于文件的客户端或者回送设备，您可以在 `/etc/modprobe.conf` 中调整回送设备数目。请编辑 `/etc/modprobe.conf` 文件并在其中添加以下行：

```
options loop max_loop=64
```

这个示例使用的是 64，但您可以使用其它数值来设定最大回送设备值。您还可以使用回送设备备份您系统中的客户端。要使用回送设备为半虚拟客户端备份客户端，请使用 `phy: block device` 命令或者 `tap:aio` 命令。要使用回送设备为全虚拟系统备份客户端，请使用 `phy: device` 或者 `file: file` 命令。

## 19.2. 在 BIOS 中启用 Intel VT 和 AMD-V 虚拟化硬件扩展

这部分论述了如何识别硬件虚拟化扩展以及如何在 BIOS 中启用它们（如果它们被禁用了）。

可在 BIOS 中禁用 Intel VT 扩展。某些笔记本电脑零售商已经在其 CPU 中默认禁用 Intel VT 扩展。

在 BIOS 中无法为在 Rev 2 套接字兼容的 AMD-V 中禁用虚拟化扩展。

有时会在 BIOS 中禁用虚拟化扩展，通常笔记本电脑生产商会这样做。启用禁用的虚拟化扩展的说明请参考 [19.2 “BIOS 中的 Intel VT 和 AMD-V 虚拟化扩展”](#)。

确定在 BIOS 中启用了虚拟化扩展。在 BIOS 中对 Intel® VT 或者 AMD-V 的设置通常是在「芯片」或者「处理器」菜单中。菜单的名称可能与本指南略有不同，虚拟化扩展设置可能位于「安全性设置」或者其它非标准菜单中。

### 过程 19.1. 在 BIOS 中启用虚拟化扩展

1. 重启计算机并打开系统 BIOS 菜单。可用使用 `delete` 或者 `Alt + F4` 组合键完成。
2. 选择「恢复默认」，然后选择「保存 & 退出」。
3. 关闭机器并断开电源。
4. 打开机器并打开「BIOS 设定工具」。打开「处理器」部分并启用 Intel® 「虚拟化技术」或者 AMD-V。在某些机器中该值还可称为「虚拟化扩展」。选择「保存 & 退出」。
5. 关闭机器并断开电源。
6. 运行 `cat /proc/cpuinfo | grep vmx svm`。如果命令有输出结果，则启用了虚拟化扩展。如果没有结果，那么您的系统可能没有虚拟化扩展或者没有启用正确的 BIOS 设置。





---

## 附录 A. 附加资源

要对 Linux 和虚拟化有更多了解，请参考下面的资源。

### A.1. 在线资源

- <http://www.cl.cam.ac.uk/research/srg/netos/xen/> Xen™ 半虚拟化机器管理器的项目网站，Fedora *kernel-xen* 软件包源于这个项目。在这个站点中保存了上级 xen 项目的二进制文件和源码，同时还保留了有关 xen 的资料、架构总览、文档和与 xen 以及其关联技术的链接。
- Xen 社区网页  
<http://www.xen.org/>
- <http://www.libvirt.org/> 是 libvirt 虚拟化 API 的官方网站。
- <http://virt-manager.et.redhat.com/> 是 Virtual Machine Manager (virt-manager) 的项目网站，它是一个管理虚拟机的图形化应用程序。
- 开源虚拟化中心  
<http://www.openvirtualization.com><sup>1</sup>
- Fedora 文档  
<http://docs.fedoraproject.org>
- 虚拟化技术总览  
<http://virt.kernelnewbies.org><sup>2</sup>
- 红帽新技术组  
<http://et.redhat.com><sup>3</sup>

### A.2. 安装的文档

- `/usr/share/doc/xen-<version-number>/` 是包含有关 Xen 半虚拟化管理程序以及相关管理工具信息的目录，其中包括各种示例配置、硬件特别信息以及当前 Xen 上级用户文档。
- `man virsh` 和 `/usr/share/doc/libvirt-<version-number>` — 包含了 `virsh` 虚拟机管理工具的子命令和选项，还包括有关 libvirt 虚拟化库 API 的综合信息。
- `/usr/share/doc/gnome-applet-vm-<version-number>` — 监控和管理本地运行的虚拟机的 GNOME 图形化面板 `applet` 的文档。
- `/usr/share/doc/libvirt-python-<version-number>` — 提供 libvirt 库的 Python 绑定的细节。libvirt-python 软件包允许 python 开发者用 libvirt 虚拟化管理库编写程序。
- `/usr/share/doc/python-virtinst-<version-number>` — 提供在虚拟机中帮助开始安装 Fedora 以及 Linux 相关发行本的 `virt-install` 命令的文档。

- `/usr/share/doc/virt-manager-<version-number>` — 提供虚拟机管理者（管理虚拟机的图形化工具）的文档。

---

## 附录 B. 修订记录

修订 12.1.3

Mon Oct 12 2009

Christopher C [ccuran@redhat.com](mailto:ccuran@redhat.com)

从红帽企业版 Linux 5.4 虚拟化指南版本 5.4-61 中分离。



---

## 附录 C. 版权

本手册书写格式为 DocBook XML v4.3。

本书主要由 Jan Mark Holzer 和 Chris Curran 完成。

其他参与者包括：

- Don Dutile -- 半虚拟驱动程序部分的技术编辑。
- Barry Donahue -- 半虚拟驱动程序部分的技术编辑。
- Rick Ring -- 虚拟机管理器部分的技术编辑。
- Michael Kearey -- 带 virsh 和虚拟软驱的 XML 配置文件应用部分的技术编辑。
- Marco Grigull -- 软件兼容性以及性能部分的技术编辑。
- Eugene Teo -- 使用 virsh 进行客户端管理部分的技术编辑。

本书的发布工具为 Publican，作者 Jeffrey Fearn。

红帽本地化团队成员组成：

### 东亚语种

- 简体中文
  - Leah Wei Liu
- 繁体中文
  - Chester Cheng
  - Terry Chuang
- 日语
  - Junko Ito
- 韩语
  - Eun-ju Kim

### 拉丁语种

- 法语
  - Sam Friedmann
- 德语
  - Hedda Peters
- 意大利语
  - Francesco Valente

- 巴西葡萄牙语
  - Glauca de Freitas
  - Leticia de Lima
- 西班牙语
  - Angela Garcia
  - Gladys Guerrero
- 俄语
  - Yuliya Poyarkova

---

# 术语表

这个词汇表是为了定义在这个安装指南中所使用的术语。

裸机	裸机指的是计算机的基本物理构架。在裸机中运行一个操作系统与在物理硬件中运行一个未修订版本的操作系统的效果一致。在裸机中运行的操作系统示例为 <i>dom0</i> ，或者是一般安装的操作系统。
dom0	还可称为 <a href="#">00</a> 或者主机操作系统。  dom0 指的是运行 <a href="#">0000</a> 的红帽企业版 Linux 主机事件，它可帮助客户端操作系统的虚拟化。Dom0 在物理硬盘中运行并管理为其自身和客户端操作系统分配物理硬件和资源。
Domain	<i>domU</i> 和 <i>Domain</i> 都是域。域在 <a href="#">0000</a> 中运行。域这个词和 <a href="#">000</a> 有类似的含义，且从技术角度来说可以互换。一个域就是一台虚拟机。
domU	domU 是指在主机系统中运行的客户端操作系统 ( <i>Domain</i> )。
全虚拟化	在全虚拟化中可以使用 Xen 和 KVM。全虚拟化使用处理器的硬件特性提供对基本物理系统 ( <a href="#">00</a> ) 的完整提取，并创建一个可供客户端操作系统运行的新虚拟系统。客户端操作系统及所有应用程序象平常一样运行，意识不到虚拟环境的存在。半虚拟化需要一个 Linux 操作系统的修订版本。
全虚拟	请参考 <a href="#">0000</a> 。
客户端系统	也称之为客户端、虚拟机或者 <i>domU</i>
硬件虚拟机	请参考 <a href="#">0000</a>
管理程序	管理程序是用来从操作系统中提取硬件信息的软件层，可允许在同一个硬件中运行多个操作系统。在主机系统中运行的管理程序还允许其它虚拟操作系统在主机硬件中运行。
主机	主机操作系统，也称为 <i>dom0</i> 。  主机操作系统环境为 <a href="#">000</a> 和 <a href="#">000</a> 运行虚拟化软件。
I/O	输入/输出 (I/O，读做“ai-ou”) 的缩写。I/O 可用来描述任意来自计算机或者向计算机传输数据的程序、操作、设备，也可用来描述来自周边设备或者向周边设备传输数据的程序、操作、设备。每次传输就是从从一个设备输出，然后输入到另一个设备中。类似键盘、鼠标等设备只能用作输入设备，而打印机则只能作为输出设备。可擦写光盘既是输入设备，也是输出设备。
基于内核的虚拟机	KVM (基于内核的虚拟机) 是为 AMD64 和 Intel64 硬件中的 Linux 准备的 <a href="#">0000</a> 解决方案。KVM 是为标准红帽企业版 Linux 内核构建的 Linux 内核模块，可运行多个未变性的虚拟客户端 Windows 和 Linux 操作系统。KVM 是使用 libvirt 虚拟化工具 ( <i>virt-manager</i> 和 <i>virsh</i> ) 的管理程序。  KVM 是用于管理程序模块本身的一组管理设备、内存和管理 API 的 Linux 内核模块。这些模块可控制作为 Linux 进程和线程运行的虚拟客户端。
LUN	逻辑单位编号 (LUN) 是为逻辑单位 (SCSI 协议条目) 分配的号码。

迁移

迁移是将一个虚拟客户端从一个主机移动到另一个主机的进程名称。迁移可离线（此时将客户端挂起然后移动）或者实时（不将客户端挂起就移动）进行。Xen 全虚拟客户端、Xen 半虚拟客户端和 KVM 全虚拟客户端都可迁移。

迁移是虚拟化的一个主要特性，因为软件是与硬件完全分开的。迁移在以下方面很有用：

- Load balancing - guests can be moved to hosts with lower usage when a host becomes overloaded.
- Hardware failover - when hardware devices on the host start to fail, guests can be safely relocated so the host can be powered down and repaired.
- Energy saving - guests can be redistributed to other hosts and host systems powered off to save energy and cut costs in low usage periods.
- Geographic migration - guests can be moved to another location for lower latency or in serious circumstances.

共享、联网存储可用来保存客户端映像。没有共享存储的迁移是无法进行的。

An offline migration suspends the guest then moves an image of the guests memory to the destination host. The guest is resumed on the destination host and the memory the guest used on the source host is freed.

离线迁移所需时间依赖网络带宽和等待时间。有 2GB 内存的客户端在 1Gbit 以太网链接中需要几秒钟。

实时迁移可让客户端继续在源主机中运行并在不停止客户端的情况下移动内存。发送映像后会追踪所有修改的内存页面并将其发送到目的地。使用更改的页面更新内存。此进程持续进行直至成功复制所有页面或者源变化太快以至目的地主机无法复制。如果出现以上两种情况，则会在源主机中暂停该客户端并发送寄存器（register）和缓存。在新主机中载入这些寄存器，然后在目的地主机中恢复该客户端。如果无法合并该客户端（客户端处于极限负载时出现），则会暂停该客户端，然后启动离线迁移。

离线迁移所需时间依赖网络带宽和等待时间以及客户端中的活动。如果该客户端正在使用大量 I/O 或者 CPU，迁移的耗时会很长。

MAC 地址

介质访问控制地址是网络接口控制器的硬件地址。在虚拟化环境中必须为虚拟网络接口生成 MAC 地址，且您本地域中的每个 MAC 都应该是唯一的。

半虚拟化

半虚拟化使用特殊的内核，有时是指 Xen 内核或者 *kernel-xen*。半虚拟客户端内核同时还使用主机程序库和设备在主机中运行。半虚拟安装可对系统中的所有设备有完全访问，但可使用安全性设置（SELinux 和文件控制）进行限制。半虚拟化比全虚拟化速度要快，可有效用于负载均衡、provisioning、安全性以及巩固性提高。

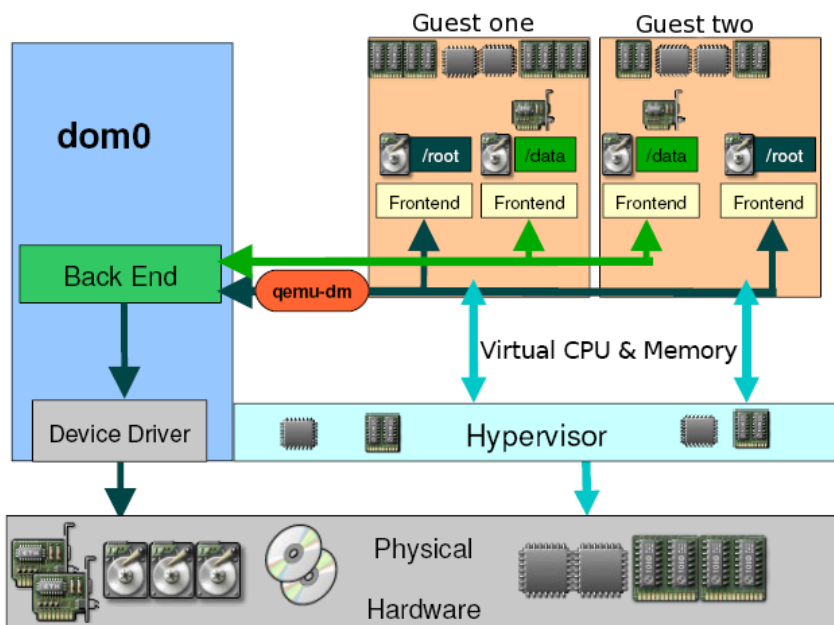


---

	<p>因为从 Fedora 9 将不再需要特殊内核。一旦将这个路径纳入主 Linux 树结构，之后的所有 Linux 版本都将会有启用的或者可用的半虚拟化系统。</p>
半虚拟	<p>请参考 <a href="#">0000</a>。</p>
半虚拟驱动程序	<p>半虚拟的驱动程序是在全虚拟 Linux 客户端中进行操作的设备驱动程序。这些驱动程序为全虚拟客户端极大提高了网络和块设备 I/O 的性能。</p>
安全性提高 Linux	<p>安全性提高 Linux 的缩写。SELinux 使用 Linux 内核中的 Linux 安全性模块 (LSM) 提供安全性策略所需的最小特权范围。</p>
全局唯一识别符	<p>全局唯一识别符 (UUID) 是对分布计算环境中的设备、系统和某些软件对象的标准化计数方法。虚拟化中的 UUID 类型包括：ext2 和 ext3 文件系统识别符、RAID 设备识别符、iSCSI 和 LUN 设备识别符、MAC 地址以及虚拟机识别符。</p>
Virtualization	<p>虚拟化是一个用于运行软件的泛泛计算术语。通常操作系统与系统中的其他程序共存且独立于其他程序。大多数目前采用的虚拟化使用管理程序。管理程序通过为客户端操作系统提供虚拟硬件，让多个操作系统可在同一物理系统中运行。虚拟化操作系统的方法有很多：</p> <ul style="list-style-type: none"> <li>• 硬件辅助虚拟化是用于 Xen 和 KVM 全虚拟化的技术（定义：<a href="#">0000</a>）</li> <li>• 半虚拟化是 Xen 用来运行 Linux 客户端的技术（定义：<a href="#">0000</a>）</li> <li>• 软件虚拟化或者模拟。软件虚拟化使用二进制转换和其它模拟技术运行未修改的操作系统。软件虚拟化比硬件辅助虚拟化或者半虚拟化要慢很多。</li> </ul>
虚拟 CPU	<p>运行红帽虚拟化的系统有很多虚拟 CPU 或者 VCPU。VCPU 的数目是限定的，并代表可分配给客户端虚拟机的 VCPU 总数。</p>
虚拟机	<p>虚拟机是物理机器或者编程语言的软件实施（例如：Java 运行时环境或者 LISP）。虚拟化环境中的虚拟机上下文就是在虚拟硬盘中运行的操作系统。</p>
Xen	<p>红帽企业版 Linux 支持 Xen 管理程序和 KVM 管理程序（请参考 <a href="#">00000000</a>）。两个管理程序有不同的构架和开发方法。Xen 管理程序在作为主机管理系统资源和虚拟化 API 的红帽企业版 Linux 操作系统中使用。主机有时指的是 <i>dom0</i> 或者 Domain0。</p>

# Xen Full Virtualization Architecture

With the para-virtualized drivers



# Xen Para-virtualization Architecture

