



# Red Hat Virtualization

## 4.0

### 技术参考

---

Red Hat Virtualization 环境的技术架构

Red Hat Virtualization Documentation Team



## Red Hat Virtualization 环境的技术架构

Red Hat Virtualization Documentation Team  
Red Hat Customer Content Services  
[rhev-docs@redhat.com](mailto:rhev-docs@redhat.com)

## 法律通告

Copyright © 2016 Red Hat.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 摘要

本参考文档介绍了 Red Hat Virtualization 环境中使用的概念、组件和技术。

# 目录

<b>第 1 章 介绍</b>	<b>3</b>
1.1. Red Hat Virtualization Manager	3
1.2. Red Hat Virtualization Host	3
1.3. Manager 的访访问接口	5
1.4. Manager 所需要的组件	6
1.5. 存储	7
1.6. 网络	8
1.7. 数据中心	9
<b>第 2 章 存储</b>	<b>11</b>
2.1. 存储域介绍	11
2.2. 组成存储域的后台存储设备类型	11
2.3. 存储域类型	11
2.4. 虚拟机磁盘镜像的存储格式	12
2.5. 虚拟机磁盘镜像存储分配策略	12
2.6. Red Hat Virtualization 的存储元数据版本	13
2.7. Red Hat Virtualization 中的存储域自动恢复	13
2.8. SPM (Storage Pool Manager)	14
2.9. SPM 选择的过程	15
2.10. Red Hat Virtualization 中的排它性资源和 Sanlock	16
2.11. 精简分配 (Thin Provisioning) 和存储过度分配 (Over-Commitment)	16
2.12. 逻辑卷扩展	17
<b>第 3 章 网络</b>	<b>18</b>
3.1. 网络架构	18
3.2. 介绍：基本网络元素	18
3.3. 网络接口控制器	18
3.4. 网桥	18
3.5. 绑定	18
3.6. 绑定的交换配置	20
3.7. 虚拟网络接口卡 (虚拟网卡)	20
3.8. 虚拟局域网 (VLAN)	21
3.9. 网络标签 (Network Label)	21
3.10. 集群网络	22
3.11. 逻辑网络	23
3.12. 必需的网络、可选网络和虚拟机网络	25
3.13. 虚拟机连接	25
3.14. 端口镜像	25
3.15. 主机网络配置	26
3.16. 网桥配置	26
3.17. VLAN 配置	27
3.18. 网桥和绑定配置	27
3.19. 多网桥、多 VLAN 和单网卡配置	28
3.20. 多网桥、多 VLAN 和单绑定配置	29
<b>第 4 章 电源管理</b>	<b>30</b>
4.1. 电源管理和隔离介绍	30
4.2. 在 Red Hat Virtualization 环境中使用代理进行电源管理	30
4.3. 电源管理	30
4.4. 隔离	31
4.5. Soft-Fencing 主机	31
4.6. 使用多个电源管理隔离代理	32

<b>第 5 章 负载均衡、调度和迁移</b>	<b>33</b>
5.1. 负载均衡、调度和迁移	33
5.2. 负载均衡策略	33
5.3. 负载均衡策略：VM_Evenly_Distributed	33
5.4. 负载均衡策略：Evenly_Distributed	33
5.5. 负载均衡策略：Power_Saving	34
5.6. 负载均衡策略：None	34
5.7. 高可用性虚拟机资源保留	34
5.8. 调度	34
5.9. 迁移	34
<b>第 6 章 目录服务</b>	<b>36</b>
6.1. 目录服务	36
6.2. 本地用户身份验证：内部域	36
6.3. 使用 GSSAPI 进行远程身份验证	36
<b>第 7 章 模板和虚拟机池</b>	<b>38</b>
7.1. 模板和虚拟机池	38
7.2. 模板	38
7.3. 虚拟机池	38
<b>第 8 章 虚拟机快照</b>	<b>40</b>
8.1. 快照	40
8.2. 实时快照	40
8.3. 创建快照	41
8.4. 预览快照	42
8.5. 删除快照	43
<b>第 9 章 硬件驱动和设备</b>	<b>44</b>
9.1. 虚拟硬件	44
9.2. 在 Red Hat Virtualization 环境中的固定设备地址	44
9.3. 中央处理器（CPU）	44
9.4. 系统设备	45
9.5. 网络设备	45
9.6. 图形设备	45
9.7. 存储设备	45
9.8. 音响设备	46
9.9. 串行驱动	46
9.10. “气球”（balloon）驱动	46
<b>第 10 章 最小的配置要求和技术限制</b>	<b>47</b>
10.1. 最小的硬件配置要求和限制	47
10.2. 数据中心的限制	47
10.3. 集群限制	47
10.4. 存储域限制	47
10.5. Red Hat Virtualization Manager 限制	48
10.6. Hypervisor 配置要求	49
10.7. 虚拟机硬件配置要求和限制	51
10.8. SPICE 的限制	52
10.9. 额外参考信息	52

## 第 1 章 介绍

### 1.1. Red Hat Virtualization Manager

Red Hat Virtualization Manager 为虚拟化环境提供了一个中央管理的功能，用户可以根据具体的情况选择使用不同的方法来访问 Red Hat Virtualization Manager。

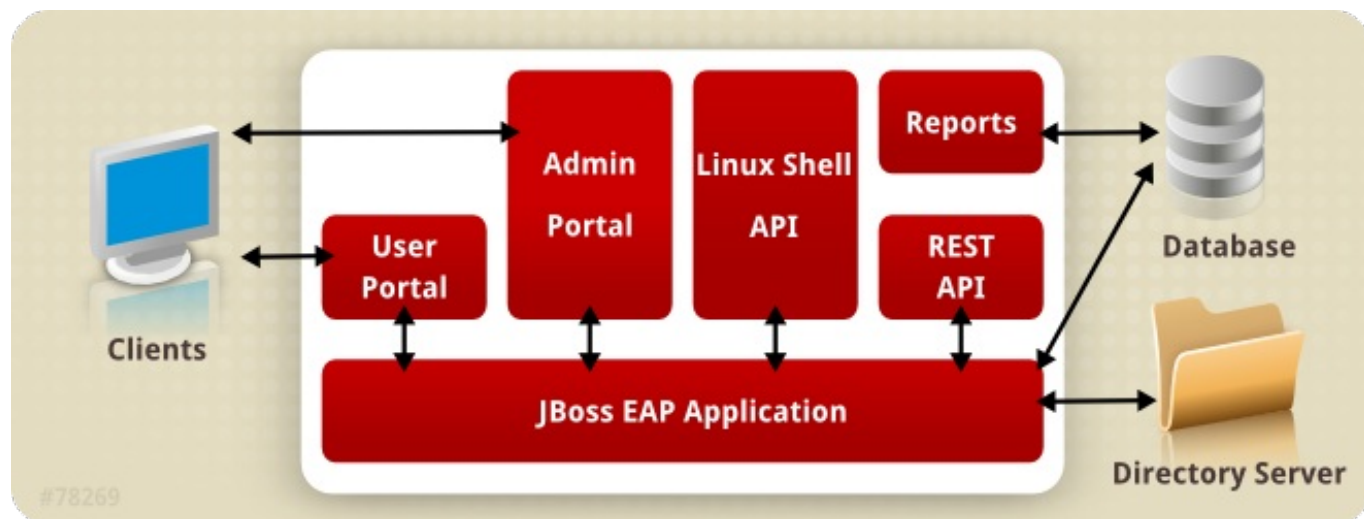


图 1.1. Red Hat Virtualization Manager 的构架

Red Hat Virtualization Manager 是一个基于内建 *Red Hat JBoss Enterprise Application Platform* 的应用程序，它为用户提供了一组图形界面接口和一组应用程序接口（API）。除了 Red Hat JBoss Enterprise Application Platform，Red Hat Virtualization Manager 还需要其它的一些组件。

### 1.2. Red Hat Virtualization Host

一个 Red Hat Virtualization 环境包括一个或多个用来运行虚拟机的主机（在本文档中，我们把它称为虚拟主机或主机）。主机为虚拟机提供运行环境的物理硬件系统。

Red Hat Virtualization Host（RHVH）运行一个专门为虚拟主机所创建的定制操作系统。

Red Hat Enterprise Linux 主机则是运行在一个标准的 Red Hat Enterprise Linux 操作系统上。在操作系统安装完成后，这个系统需要做进一步的配置，来使它可以作为一个运行虚拟机的主机。

以上两种形式的主机都以相同的方式和虚拟环境中的其它项进行交流，因此我们把它们都统称为主机。

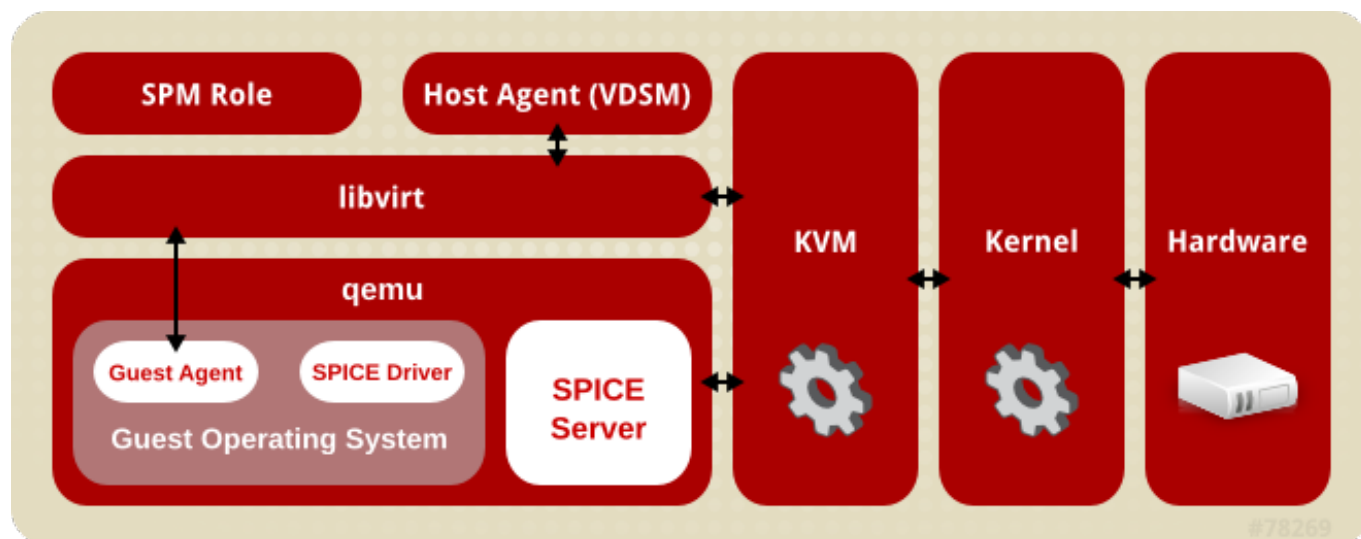


图 1.2. 主机的构架

## Kernel-based Virtual Machine (KVM)

Kernel-based Virtual Machine (KVM) 是一个可加载的内核模块，它通过使用 Intel VT 或 AMD-V 硬件扩展来提供虚拟化的功能。KVM 本身运行于内核空间，而在它上面运行的虚拟机服务会作为独立的 QEMU 进程在用户空间中运行。KVM 允许主机把它的物理硬件资源分配给虚拟机。

## QEMU

QEMU 是一个多平台的、提供全仿真功能的“仿真器(emulator)”，它会仿真包括一个或多个处理器以及外设在内的整个系统（如一台 PC）。QEMU 可以被用来运行不同的操作系统，或用来进行系统代码故障排除。QEMU、KVM 和带有适当虚拟化扩展功能的处理器组合在一起可以提供虚拟化功能。

## Red Hat Virtualization Manager 主机代理 - VDSM

在 Red Hat Virtualization 环境中，VDSM 被用来启动虚拟机和存储设备上的操作。另外，它还被用来处理不同主机间的通讯。VDSM 会监控主机的资源（如内存、存储和网络），并管理各种任务，如虚拟机创建、数据统计和日志收集等。每个主机上都会运行一个 VDSM 服务来通过端口 **54321**（这个端口值可以被重新配置）接收 Red Hat Virtualization Manager 所发出的管理命令。

## VDSM-REG

VDSM 使用 VDSM-REG 在 Red Hat Virtualization Manager 上注册所有主机。VDSM-REG 需要使用端口 **80** 或 **443** 来提供它本身的信息，以及主机的信息。

## libvirt

libvirt 被用来协调虚拟机以及和虚拟机相关的虚拟设备的管理。当 Red Hat Virtualization Manager 发出一个操作虚拟机的命令时（如启动虚拟机、停止虚拟机、重启虚拟机），VDSM 会调用相关主机上的 libvirt 来执行这个命令。

## Storage Pool Manager (SPM)

Storage Pool Manager (SPM) 是分配给一个数据中心中的某个主机的角色。作为 SPM 的主机（我们称它为 SPM 主机）全权负责数据中心中的所有与存储域相关的变化（如创建、删除和修改虚拟磁盘镜像、快照和模板）。同时，它还负责为在 Storage Area Network (SAN) 中的稀疏块设备分配存储资源的任务。SPM 的角色可以在同一个数据中心的不同的主机间进行迁移，因此同一个数据中心中的所有主机都必须可以访问在这个数据中心中定义的所有存储域。

Red Hat Virtualization Manager 会确保 SPM 一直处于有效的状态。如果 SPM 所在的主机出现问



题，Manager 会把 SPM 角色分配给数据中心中的另外一个主机。

## 虚拟机操作系统

操作系统以及在它上面运行的应用程序可以在不进行任何修改的情况下在 Red Hat Virtualization 环境中的虚拟机上安装并运行，它们与运行在物理机器上的程序没有任何区别。

红帽还提供了一组增强的设备驱动程序，使用它们可以更块、更有效地访问虚拟设备。另外，您还可以在虚拟机上安装 Red Hat Virtualization Guest Agent，它可以为管理控制台提供更多的虚拟机信息。

## 1.3. Manager 的访问接口

### 用户门户 (User Portal)

桌面系统虚拟化 (desktop virtualization) 为用户提供了和一个 PC 桌面系统相似的桌面系统。用户可以使用网络浏览器来访问用户门户 (User Portal)，并通过用户门户来获得分配给他们的 *虚拟桌面资源*。系统管理员需要为每个用户设定他们可以获得的资源。标准用户可以启动、停止和使用分配给他们的虚拟桌面系统，而高级用户则可以执行一些管理任务。标准用户和高级用户都使用相同的 URL 来访问用户门户，系统会根据登录时所使用的用户帐号来决定用户可以执行哪些操作。

#### ✦ 标准用户

标准用户可以通过用户门户来启动或关闭相应的虚拟机。另外，用户也可以使用 SPICE (*Simple Protocol for Independent Computing Environments*) 或 VNC (*Virtual Network Computing*) 协议来直接连接到相应的虚拟机。系统管理员会在创建虚拟机时指定用户可以使用哪种协议来直接连接到虚拟机。

如需了解更多与用户门户相关的信息，请参阅 [Introduction to the User Portal](#)。

#### ✦ 高级用户

Red Hat Virtualization 用户门户为高级用户提供了一个界面来创建、使用和监控虚拟机资源。系统管理员还会把一些可以执行管理任务的权限分配给高级用户。除了标准用户可以执行的任务外，高级用户通常还可以执行以下任务：

- 创建、编辑和删除虚拟机。
- 管理虚拟磁盘和网络接口。
- 为用户分配虚拟机的权限。
- 创建和使用模板来快速部署虚拟机。
- 监测资源的使用情况和相关的系统事件。
- 创建和使用快照来进行虚拟机的恢复。

虽然高级用户可以执行虚拟机的管理任务，但是数据中心和集群一级的管理任务只能由系统管理员执行。

### 管理门户 (Administration Portal)

管理门户是 Red Hat Virtualization Manager 服务器的一个图形接口。管理员使用网络浏览器通过它来监控、创建并维护虚拟环境中的所有资源。通过管理门户可以执行以下操作：

- ✦ 创建和管理虚拟基础架构（网络、存储域）。
- ✦ 安装和管理主机。

- ✦ 创建和管理逻辑项（数据中心、集群）。
- ✦ 创建和管理虚拟机。
- ✦ Red Hat Virtualization 用户和权限管理。

管理门户需要使用 JavaScript。

[Red Hat Virtualization 管理指南](#)中提供了更详细的关于使用管理门户的信息。如需了解管理门户所支持的浏览器和平台信息，请参阅 [Red Hat Virtualization 安装指南](#)。

## Representational State Transfer (REST) API

Red Hat Virtualization REST API 提供了一个用来集成和控制 Red Hat Virtualization 环境的软件接口。用户可以使用任何支持 HTTP 操作的编程语言来使用 REST API。

使用 REST API 可以：

- ✦ 把虚拟环境集成到 IT 环境中。
- ✦ 与第三方虚拟化软件进行集成。
- ✦ 自动化维护和错误检查任务。
- ✦ 使用脚本在 Red Hat Virtualization 环境中执行重复性的操作。

请参阅 [Red Hat Virtualization REST API 指南](#)来获得与 API 相关的信息以及相应的实例。

## 1.4. Manager 所需要的组件

### Red Hat JBoss Enterprise Application Platform

Red Hat JBoss Enterprise Application Platform 是一个 Java 应用服务器，它为跨平台的 Java 应用程序开发和部署提供了一个构架。Red Hat Virtualization Manager 是通过使用 Red Hat JBoss Enterprise Application Platform 来部署的。



#### 重要

包括在 Red Hat Virtualization Manager 中的 Red Hat JBoss Enterprise Application Platform 是为运行 Red Hat Virtualization Manager 特殊定制的，因此不能为其它应用程序提供服务。如果使用 Manager 中的 Red Hat JBoss Enterprise Application Platform 用于其它目的，则会对 Red Hat Virtualization 环境造成影响。

### 收集报表和历史数据

Red Hat Virtualization Manager 包括了一个数据仓库，它被用来收集监控主机、虚拟机和存储所产生的数据。Red Hat Enterprise Virtualization Manager 提供了一组预先定义的报表，用户也可以使用任何支持 SQL 查询的工具来创建报表。

Red Hat Virtualization Manager 会在安装的过程中，在指定的 Postgres 数据库服务器上创建两个数据库。

- ✦ engine 数据库是 Red Hat Virtualization Manager 用来保存数据的主数据库，它保存了与虚拟环境相关的数据（如虚拟环境的状态、配置和性能数据）。

- `ovirt_engine_history` 数据库保存了配置信息和各种性能统计数据，这些数据是根据不同时间从 `engine` 数据库中收集来的数据所产生的。系统在每一分钟都会检查 `engine` 数据库中的配置信息，任何改变都会被记录在 `ovirt_engine_history` 数据库中。记录在这个数据库中的历史信息可以被用来帮助您分析和提高 Red Hat Virtualization 环境的性能，或帮助您解决存在的问题。

如需了解更多与使用 `ovirt_engine_history` 数据库产生报表的信息，请参阅 *Red Hat Virtualization Data Warehouse Guide* 中的 [History Database](#)。



### 重要

在 `ovirt_engine_history` 数据库中记录数据的功能是通过 **RHEVM History Service** (`ovirt-engine-dwhd`) 实现的。

## 目录服务

目录服务提供了一个基于网络的、中央存储的用户和机构信息。这些信息包括应用程序的设置、用户档案、组数据、策略信息和访问控制信息。Red Hat Virtualization Manager 支持 Active Directory、Identity Management (IdM)、OpenLDAP 和 Red Hat Directory Server 9 所提供的目录服务。另外，它还包括了一个本地的内部域，这个域只被用来进行系统管理，并只包括一个用户 - `admin`。

## 1.5. 存储

Red Hat Virtualization 使用一个中央化的存储系统来保存虚拟机的磁盘镜像、模板、快照和 ISO 文件。所有存储被分为不同的逻辑组（称为存储池），而这些存储池组成了存储域。存储域由一组存储空间和代表这些存储空间内部结构的元数据所组成，它被分为 3 类：数据存储域、导出存储域和 ISO 存储域。

每个数据中心都必须包括一个数据存储域（导出存储域和 ISO 存储域则不是必需的），而每个数据存储域也只对一个数据中心有效。存储域是用来提供共享资源的，因此它必须可以被所在数据中心中的所有主机访问。

存储网络可以通过使用 Network File System (NFS)、Internet Small Computer System Interface (iSCSI)、GlusterFS、Fibre Channel Protocol (FCP) 或任何与 POSIX 兼容的网络文件系统实现。

在 NFS（以及其它 POSIX 兼容的文件系统）域中，所有的虚拟磁盘、模板和快照都以文件的形式来代表。

在 SAN (iSCSI/FCP) 域中，不同的块设备被 Logical Volume Manager (LVM) 组成为一个卷组 (Volume Group - VG)，每个虚拟磁盘、模板和快照都是 VG 中的一个逻辑卷 (Logical Volume - LV)。请参阅 [Red Hat Enterprise Linux Logical Volume Manager Administration Guide](#) 来获得更多关于 LVM 的信息。

### 数据存储域

数据域保存了在环境中运行的所有虚拟机上的虚拟硬盘镜像、模板和快照。一个数据域不能被不同的数据中心所共享。

### 导出存储域

导出域是一个临时的数据存储仓库，它被用来在数据中心和 Red Hat Virtualization 环境间复制和迁移数据镜像。导出域可以被用来备份虚拟机和模板。一个导出域可以在不同的数据中心间迁移，但它只能同时在一个数据中心中有效。

### ISO 存储域

ISO 域保持了用来在虚拟机上安装操作系统和应用程序的逻辑 CD-ROM 的 ISO 文件。通过使用 ISO 域中的 ISO 文件，数据中心将不再需要物理的安装介质。一个 ISO 域可以被不同数据中心共享。

## 1.6. 网络

Red Hat Virtualization 网络架构被用来处理 Red Hat Virtualization 环境中的不同对象间的网络连接。除此之外，它还被用来实现网络的隔离。

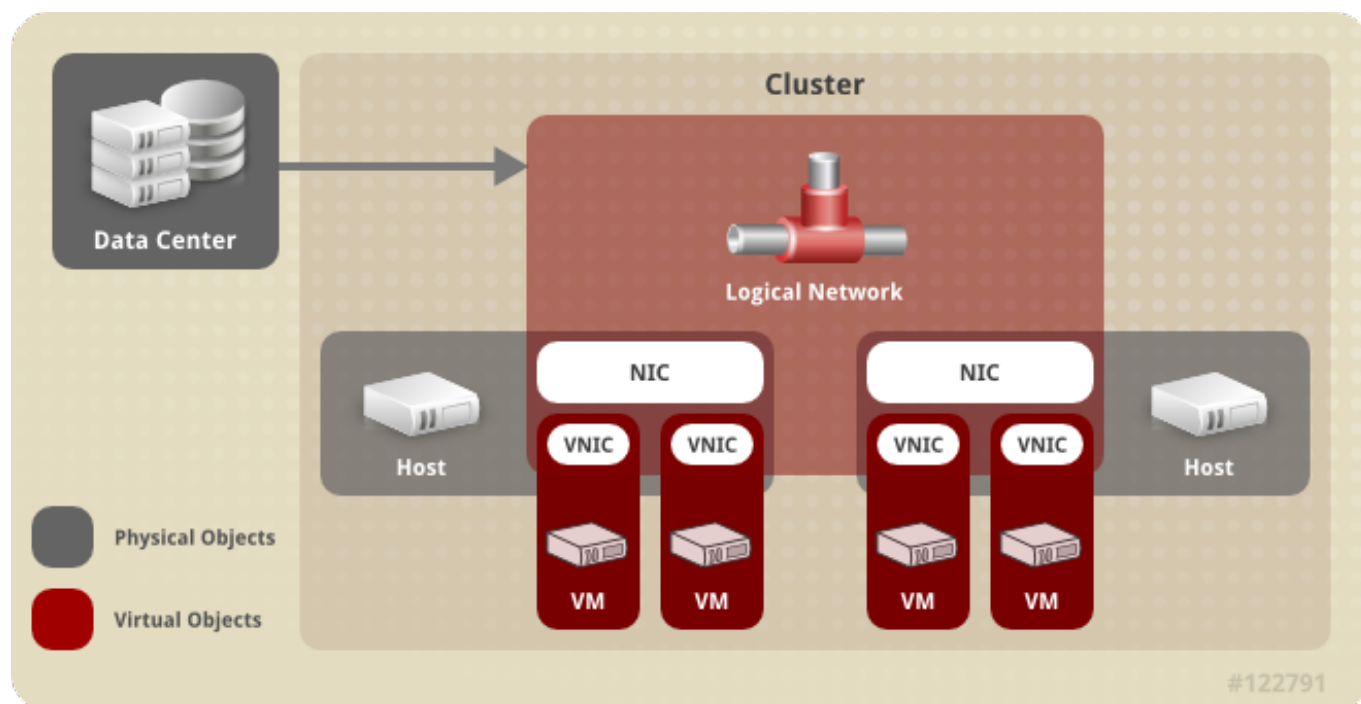


图 1.3. 网络架构

网络需要在 Red Hat Virtualization 的不同层上被定义，而支持这些网络功能的物理网络架构需要被配置来实现硬件和 Red Hat Virtualization 环境中的逻辑组件间的网络连接。

### 网络架构层

Red Hat Virtualization 网络架构需要以下的硬件和软件设备：

- ✧ 网络接口控制卡（简称网卡或 NIC）是一个物理网络接口设备，用来把主机连接到网络中。
- ✧ 虚拟网卡（VNIC）是一个逻辑 NIC。它利用主机的物理网卡来实现虚拟机间的网络连接。
- ✧ 绑定（bond）是由多个网卡组成的一个网络接口。
- ✧ 网桥（bridge）是一个使用数据包转发技术实现的数据包交换网络。它是实现虚拟机逻辑网络的基础。

### 逻辑网络

使用逻辑网络可以根据环境的要求对网络流量进行分隔。逻辑网络有以下几种类型：

- ✧ 处理虚拟机网络流量的逻辑网络，
- ✧ 不处理虚拟机网络流量的逻辑网络，
- ✧ 可选的逻辑网络，
- ✧ 必需的逻辑网络。

所有逻辑网络都可以是“必需的”逻辑网络或“可选的”逻辑网络。

用来处理虚拟机间网络连接的逻辑网络是通过主机层上的基于软件的网桥设备实现的。在默认情况下，Red Hat Virtualization Manager 会在它的安装过程中创建一个逻辑网络：**ovirtmgmt** 管理网络。

另外，系统管理员还可以添加专用的存储逻辑网络和专用的显示逻辑网络。那些不需要处理虚拟机网络流量的逻辑网络将不会在主机上有相应的网桥设备，它们与主机上的网络接口直接相关联。

Red Hat Virtualization 把管理相关的网络流量和迁移相关的网络流量隔离。这将允许使用一个专用的网络（不需要路由）来处理虚拟机迁移所产生的网络流量，从而保证管理网络（ovirtmgmt）在虚拟机迁移的过程中可以正常运行。

## 不同层上的逻辑网络介绍

逻辑网络对于虚拟环境中的不同层有不同的意义。

### 数据中心层

逻辑网络在数据中心层上被定义。在默认情况下，每个数据中心都有一个 **ovirtmgmt** 管理网络，用户还可以添加其它的逻辑网络。为数据中心所定义的逻辑网络需要被添加到使用它们的集群中。

### 集群层

在数据中心层上定义的逻辑网络需要被添加到使用它们的集群中。在默认的情况下，每个集群都会连接到管理网络中，您还可以把集群所在数据中心中的逻辑网络添加到集群中。当一个“必需的”逻辑网络被添加到集群中时，它需要被添加到集群中的所有主机上；而一个“可选的”逻辑网络可以根据需要被添加到所需的主机上。

### 主机层

虚拟机逻辑网络是通过集群中的每个主机上的一个软件网桥设备实现的，这个软件网桥和一个特定的网络接口相关联；而非虚拟机逻辑网络不和任何网桥相关联，它直接和主机上的网络接口相关联。Red Hat Virtualization 环境中所包括的管理网络是通过和主机上的某个网络设备相关联的网桥实现的；而添加到集群中的其它逻辑网络需要和集群中的某个主机的网络接口直接相关联。

### 虚拟机层

在虚拟机上使用逻辑网络的方式和在物理机器上使用网络的方式相同。虚拟机可以使用一个虚拟网络连接运行它的主机上的逻辑网络中，并可以使用所连接虚拟网络中的资源。

#### 例 1.1. 管理网络

管理逻辑网络（**ovirtmgmt**）在安装 Red Hat Virtualization Manager 的过程中被自动创建，它专门用来管理 Red Hat Virtualization Manager 和主机间的网络数据。如果没有设置其它的网桥，所有的网络数据将使用 **ovirtmgmt** 作为默认的网桥。

## 1.7. 数据中心

数据中心是 Red Hat Virtualization 环境中的最高一级的项，它包括以下三个子项（子容器）：

- ✱ **存储容器**用来保存存储类型和存储域的信息，以及存储域间的连接信息。存储在数据中心一级上定义，并可以被所在数据中心中的所有集群使用。
- ✱ **网络容器**用来保存与数据中心中的逻辑网络相关的信息，包括网络地址、VLAN 标签（tag）和 STP 支持等信息。逻辑网络在数据中心一级上定义，并可以在集群一级上使用。

- ✱ **集群容器**用来保存与集群相关的信息。集群就是一组有兼容处理器内核（AMD 或 Intel）的主机。集群组成了一个虚拟机迁移域，虚拟机可以被实时迁移到所在集群中的其它主机上（但不能迁移到其它集群的主机上）。一个数据中心可以包括多个集群，一个集群可以包括多个主机。



## 第 2 章 存储

### 2.1. 存储域介绍

存储域就是一组有一个公共存储接口的数据镜像，它包括了模板、虚拟机（包括快照）的数据镜像或 ISO 文件以及存储域本身的元数据。一个存储域可以由块设备（SAN -- iSCSI 或 FCP）组成，也可以由文件系统（NAS -- NFS、GlusterFS，或其它 POSIX 兼容的文件系统）组成。

在 NFS 中，所有的磁盘、模板和快照都是文件。

在 SAN (iSCSI/FCP) 中，每个虚拟磁盘、模板和快照都是一个逻辑卷。块设备被组合到一个逻辑卷组中，并被逻辑卷管理器（Logical Volume Manager，简称 LVM）分为不同的逻辑卷作为虚拟硬盘供用户使用。如需了解更多与 LVM 相关的信息，请参阅 [Red Hat Enterprise Linux Logical Volume Manager Administration Guide](#)。

逻辑硬盘可以有两种格式：QCOW2 或 RAW，存储类型可以是 Sparse 或 Preallocated。快照的类型是 sparse，但它可以是为 RAW 或 sparse 磁盘创建的。

共享相同存储域的虚拟机可以在同一个集群中的主机间进行迁移。

### 2.2. 组成存储域的后台存储设备类型

存储域可以使用基于块的存储设备，也可以使用基于文件的存储设备。

#### 基于文件的存储

Red Hat Virtualization 支持的基于文件的存储类型包括：NFS、GlusterFS、其它 POSIX 兼容的文件系统以及主机的本地存储。

一般情况下，Red Hat Virtualization 环境并不管理基于文件的存储。

NFS 存储由 Red Hat Enterprise Linux NFS 服务器，或其它第三方的网络存储服务器来管理。

主机可以管理它们自己的本地文件存储系统。

#### 基于块的存储

块存储使用没有格式化的块设备。块设备被 Logical Volume Manager (LVM) 分为卷组，VDSM 会通过扫描卷组的变化来在 LVM 之上添加集群的逻辑。当 VDSM 发现卷组变化时，它会通知相应的主机来更新它们的卷组信息。主机会把卷组分为不同的逻辑卷，并在磁盘上保存逻辑卷的元数据。当在已经存在的存储域中添加新的存储空间时，Red Hat Virtualization Manager 会通知每台主机上的 VDSM 来更新卷组的信息。

Logical Unit Number (LUN) 是一个独立的块设备，它会通过块存储协议（iSCSI、FCoE 或 SAS）进行连接。Red Hat Virtualization Manager 只管理使用软件 iSCSI 到 LUN 的连接，而使用其它块存储协议到 LUN 的连接由 Red Hat Virtualization 环境外的系统所管理。一个被选择作为 Storage Pool Manager (SPM) 的主机上的 LVM 会处理基于块的存储环境中发生的任何变化（如创建逻辑卷、扩展或删除逻辑卷、添加新的 LUN），然后 VDSM 会更新集群中的所有主机上的元数据来和所发生的变化进行同步。

### 2.3. 存储域类型

Red Hat Virtualization 支持的存储域可以被分为以下几类：

- ✱ **数据存储域**：保存 Red Hat Virtualization 环境中的所有虚拟机的磁盘镜像。这些磁盘镜像会包括安装的操作系统，或由虚拟机产生或保存的数据。数据存储域支持 NFS、iSCSI、FCP、GlusterFS 或 POSIX 兼容

的存储系统。一个数据存储域不能在不同的数据中心间共享。

- ✱ **导出存储域**：为在不同数据中心间转移磁盘镜像和虚拟机模板提供一个中间存储，并可以用来保存虚拟机的备份。导出存储域支持 NFS 存储。一个导出域可以被多个不同的数据中心访问，但它只能同时被一个数据中心使用。
- ✱ **ISO 存储域**：用来存储 ISO 文件（也称为镜像，它是物理的 CD 或 DVD 的代表）。在 Red Hat Virtualization 环境中，ISO 文件通常代表了操作系统的安装介质、应用程序安装介质和 guest 代理安装介质。这些镜像会被附加到虚拟机上，并象使用物理安装介质一样启动系统。ISO 存储域为数据中心中的所有主机提供了一组共享的 ISO 文件，这样所有的主机就不再需要物理的安装介质了。

## 2.4. 虚拟机磁盘镜像的存储格式

### 使用 QCOW2 格式的虚拟机存储

QCOW2（QCOW 是 *QEMU copy on write* 的缩写）是虚拟机磁盘镜像的一种存储格式，使用 QCOW2 格式可以把物理存储层和逻辑存储层分隔开。QCOW2 为逻辑块和物理块之间创建了一个映射信息，每个逻辑块都会被映射到相应的物理块上；另外，QCOW2 可以只保存物理存储上的数据变化。因为这些特性，存储空间“过度分配（over-commitment）”功能和虚拟机快照功能才能得以实现。

初始的映射信息会把所有的逻辑块与物理文件系统或卷中对应的块相关联。在创建虚拟机快照后，如果这个虚拟机需要向 QCOW2 卷写数据，系统会根据映射信息在物理存储中找到相应的块，并把新数据写到块中，然后只在新的快照 QCOW2 卷中记录数据的变化，并更新相应的映射信息。

### RAW

当虚拟机磁盘的镜像为 RAW 格式时，它上面的数据将没有特定的格式，对虚拟磁盘的操作也不需要主机进行特殊处理，因此使用 RAW 格式的虚拟机磁盘会比使用 QCOW2 格式的虚拟磁盘有更好的性能。当虚拟机向虚拟磁盘写数据时，I/O 系统会在物理存储和逻辑卷中写相同的数据。

除非使用由外部存储阵列所管理的“自动精简配置（Thin Provisioned）”LUN，RAW 格式的虚拟磁盘需要在创建时就被分配和所定义的镜像大小相同的存储空间。

## 2.5. 虚拟机磁盘镜像存储分配策略

### 预分配存储（Preallocated Storage）

虚拟机磁盘镜像所需要的所有存储空间在虚拟机创建前就需要被完全分配。如果虚拟机需要一个 20 GB 的磁盘镜像，存储域中的 20GB 的存储空间就需要被占用。因为在进行写操作时不需要分配磁盘空间，所以预分配存储有更好的写性能。但是，预分配存储的大小不能被扩展，这就失去了一些灵活性。另外，它也会降低 Red Hat Virtualization Manager 进行存储“过度分配”的能力。预分配存储适用于需要大量 I/O 操作，并对存储速率有较高要求的虚拟机，一般情况下，虚拟服务器适于使用预分配存储。



#### 注意

如果您的后台存储设备提供了精简分配（thin provisioning）功能，在您通过管理门户为虚拟机分配存储时，仍然需要选择预分配存储。

### 稀疏分配存储（Sparsely Allocated Storage）

在创建虚拟机的时候，为虚拟机磁盘镜像设定一个存储空间上限，而磁盘镜像在开始时并不使用任何存储域中的存储空间。当虚拟机需要向磁盘写数据时，磁盘会从存储域中获得存储空间，直到



达到了创建时所设置的磁盘空间上限。当数据从磁盘镜像中被删除后，空余的存储空间不会被返回到存储域中。稀疏分配存储适用于不需要大量 I/O 操作，并对存储性能要求不高的系统。一般情况下，它适用于桌面虚拟机。



### 注意

如果您的后台存储设备提供了精简分配（thin provisioning）功能，您应该首选使用后台设备所提供的功能来实现精简配置。在您通过图形用户界面为虚拟机配置存储时，选择预分配存储，后台存储会实现精简配置的功能。

## 2.6. Red Hat Virtualization 的存储元数据版本

Red Hat Virtualization 把存储域的信息作为元数据存储存储在存储域中。每个新的 Red Hat Virtualization 版本都包括对存储元数据实现的改进。

### ✧ V1 元数据 (Red Hat Virtualization 2.x 系列)

每个存储域的元数据包括了存储域本身的结构，以及所有被虚拟机磁盘镜像使用的物理卷的名字。

主域的元数据还额外包括了存储池中的所有域和物理卷的名字。因为这个元数据的大小不能超过 2 KB，所以它限制了一个池中所能包括的存储域的数量。

模板和虚拟机的基本数据镜像是只读的。

V1 元数据适用于 NFS、iSCSI 和 FC 存储域。

### ✧ V2 元数据 (Red Hat Enterprise Virtualization 3.0)

所有存储域和池的元数据以逻辑卷标签的形式保存（不再被写到一个逻辑卷上）。而虚拟机磁盘卷的元数据仍然以一个逻辑卷的形式保存在存储域中。

元数据将不再包括物理卷名。

模板和虚拟机的基本数据镜像是只读的。

V2 元数据适用于 iSCSI 和 FC 存储域。

### ✧ V3 元数据 (Red Hat Enterprise Virtualization 3.1+)

所有存储域和池的元数据以逻辑卷标签的形式保存（不再被写到一个逻辑卷上）。而虚拟机磁盘卷的元数据仍然以一个逻辑卷的形式保存在存储域中。

虚拟机和模板的基本镜像数据不再是只读的了。这使实时快照、实时存储迁移和快照克隆成为可能。

支持 unicode 元数据。它可以被用来支持非英文的卷名。

V3 元数据适用于 NFS、GlusterFS、POSIX、iSCSI 和 FC 存储域。

## 2.7. Red Hat Virtualization 中的存储域自动恢复

Red Hat Virtualization 环境中的主机会通过读所在数据中心中的存储域元数据来监测存储域。当一个数据中心中的所有主机都报告某个存储域无法访问时，这个存储域就被认为“不活跃”。

Manager 不会在某个存储域不活跃时取消它的连接，而是假设它是由一个临时的网络故障造成的。Manager 会每 5 分钟尝试重新激活任何“不活跃”的存储域。

在这种情况下，虽然系统管理员可能需要手工排除造成存储域连接出现问题的故障，但是 Manager 会在连接问题解决后自动重新激活存储域。

## 2.8. SPM (Storage Pool Manager)

Red Hat Virtualization 使用元数据来描述存储域的内部结构。结构元数据会被写到每个存储域的一个数据段中，它被用来记录镜像和快照的创建和删除操作，以及卷和域的扩展操作。所有主机会使用“一人写，多人读”的机制来处理存储域元数据。

可以对数据域的结构进行改变的主机称为 SPM (Storage Pool Manager)，它会协调数据中心中的所有存储域元数据的改变（如创建和删除磁盘镜像、创建和迁移快照、在存储域间复制镜像、创建模板和为块设备分配存储）。每个数据中心只能有一个主机作为 SPM，其它的主机只能读存储域的结构元数据。

一个主机可以被手动指定为 SPM，也可以被 Red Hat Virtualization Manager 自动指定。当 Manager 指定一个主机作为 SPM 时，它会试图使主机获得一个名为 *存储为中心的租约* (*storage-centric lease*)，这个租约将允许 SPM 主机写存储元数据。“存储为中心”意味着它会直接写存储域，而不需要 Manager 或主机对它进行控制。存储为中心的租约会被写到主存储域中的名为 **leases** 的一个特殊逻辑卷上，而关于存储域结构的元数据会被写到名为 **metadata** 的一个特殊逻辑卷上。修改 **metadata** 逻辑卷的操作会受到 **leases** 逻辑卷的保护。

Manager 使用 VDSM 向主机发一个 **spmStart** 命令，主机上的 VDSM 在接到命令后会试图使主机获得“存储为中心的租约”。如果主机成功获得这个租约，它将成为 SPM。在 Red Hat Virtualization Manager 指定另外一台主机作为 SPM 前，这个主机会一直保持这个租约。

当以下情况发生时，Manager 会指定另外一个主机作为 SPM：

- ✱ SPM 主机不能访问所有存储域，但可以访问主存储域。
- ✱ SPM 主机无法对“存储为中心的租约”进行续约（因为存储连接断开，或 lease 卷没有可用空间连进行写操作）。
- ✱ SPM 主机出现故障。

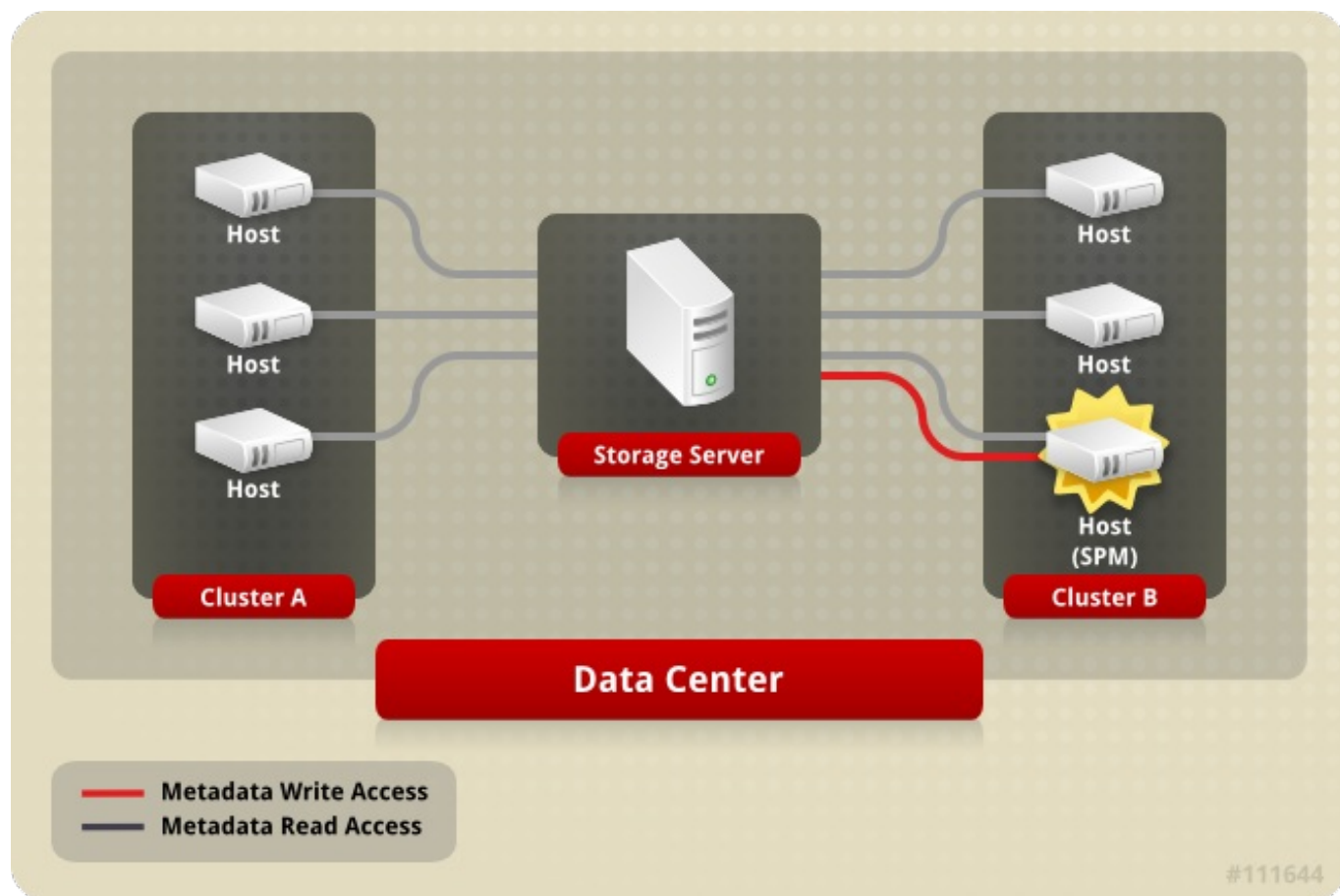


图 2.1. 只有 SPM 可以写结构元数据。

## 2.9. SPM 选择的过程

如果没有主机被手工指定为 SPM，Red Hat Virtualization Manager 将会启动并管理 SPM 的选择过程。

首先，Red Hat Virtualization Manager 会要求 VDSM 确认哪个主机已经有了“存储为中心的租约”。

Red Hat Virtualization Manager 会跟踪从存储域创建以来的 SPM 分配记录。主机是否可以作为 SPM 由以下 3 方面决定：

- ✧ "getSPMstatus" 命令：Manager 使用 VDSM 检查主机的 SPM 状态。它会返回以下 3 个状态之一："SPM"、"Contending" 或 "Free"。
- ✧ 存储域的元数据卷中包括了最后一个具有 SPM 身份的主机信息。
- ✧ 存储域的元数据卷中包括了最后一个具有 SPM 身份的主机版本信息。

如果当前的 SPM 可以正常工作，这个主机会继续保持“存储为中央的租约”，Red Hat Virtualization Manager 在管理门户中把那台主机标识为 SPM，并不再进行以下操作。

如果当前的 SPM 主机没有响应，它被认为处于“无响应”状态。如果那个主机已经配置了电源管理功能，它会被自动隔离（fence）。如果自动隔离失败，就需要手工隔离。在当前的 SPM 被隔离前，SPM 的角色不能分配给其它主机。

当 SPM 角色和存储为中心的租约都空闲时，Red Hat Virtualization Manager 会把它们分配给数据中心中的一个随机选取的主机。

如果为一个新主机分配 SPM 角色的操作失败，Red Hat Virtualization Manager 会把这个主机加入到一个包括了所有被分配 SPM 角色失败的主机列表中（这个列表中的主机被标记为无法成为 SPM）。这个列表中的内容会在开始下一次进行 SPM 选择的过程前被清除，从而使这个列表中的主机又有机会成为 SPM。

Red Hat Virtualization Manager 会一直尝试把 SPM 角色和存储为中央的租约分配给一个主机，直到有一个主机成功成为了 SPM。

在当前的 SPM 没有响应或无法完成正常的任务时，Red Hat Virtualization Manager 就会启动 SPM 选择的过程。

## 2.10. Red Hat Virtualization 中的排它性资源和 Sanlock

Red Hat Virtualization 环境中的一些资源具有排它性，每个排它性资源只能同时被一个对象访问。

SPM 就是一个具有排它性的资源。在一个数据中心中，只能同时有一个主机具有 SPM 角色，如果数据中心中存在多个具有 SPM 角色的主机，就会出现同一个数据同时被不同的主机进行修改的情况，从而造成数据被破坏。

在 Red Hat Enterprise Virtualization 3.1 之前，SPM 的排它性是通过 VDSM 中的一个名为 *safelease* 的功能来实现的。这个租约被写到数据中心中的所有存储域中的一个特殊区域中，而数据中心中的所有主机都可以通过它来检查 SPM 的状态。VDSM 的“*safelease*”的唯一功能就是来维持 SPM 的排它性。

Sanlock 可以通过“锁定 (lock)” SPM 角色来提供相同的功能，但它还可以“锁定”其它资源。因此，Sanlock 具有更高的灵活性。

需要进行资源锁定的应用程序可以在 Sanlock 中进行注册，已经注册的应用程序可以请求 Sanlock 锁定某个资源，从而使其它程序无法访问被锁定的资源。例如，VDSM 可以请求 Sanlock 锁定 SPM 资源，而不需要自己锁定它。

每个存储域都有一个 *lockspace* 区，锁定状态被记录在 *lockspace* 的磁盘中。因为 SPM 资源只能分配给一个“活跃的”主机，因此 Sanlock 就需要检查具有 SPM 的主机是否是“活跃的”。当 SPM 主机连接到存储域时，它会更新从 Manager 获得的 *hostid*，并且会定期在 *lockspace* 中写一个时间戳 (timestamp)。ids 逻辑卷会记录每个主机的 ID，并在每个主机更新它的 *hostid* 时进行相应的更新。Sanlock 会根据主机的 *hostid* 刷新和时间戳来决定它是否处于“活跃”状态。

资源的使用情况被记录在 *leases* 逻辑卷的磁盘中。当磁盘中代表某个资源的数据被更新为带有某个进程的 id 时，系统就认为这个资源被这个进程所占用。具体到 SPM 角色资源，当它被占用时，它的数据会被更新为带有成为 SPM 的主机的 *hostid*。

每个主机上的 Sanlock 只需要检查资源一次来决定它们是否被占用。在初始的检查后，Sanlock 只需要监测 *lockspaces* 中的相应主机的时间戳的状态。

Sanlock 需要监测使用资源的应用程序。对于 VDSM，它会监测 SPM 的状态和 *hostid*。如果主机无法从 Manager 重复获得它的 *hostid*，这个主机就会失去它所占有的、在 *lockspace* 中记录的所有资源的排它性。Sanlock 会更新相应的资源记录来标识这些资源不再被占用。

当 SPM 主机在一定的时间内无法在存储域的 *lockspace* 中写时间戳时，这个主机上的 Sanlock 会要求 VDSM 进程释放它所占用的资源。如果 VDSM 进程接受了这个请求，它将释放它所占用的资源，*lockspace* 中的 SPM 资源就可以被其它主机使用。

如果 SPM 主机上的 VDSM 无法接受释放资源的请求，主机上的 Sanlock 就会使用 *kill* 命令来终止 VDSM 进程。如果 *kill* 命令运行失败，Sanlock 会使用 *sigkill* 命令来终止 VDSM 进程。如果 *sigkill* 命令仍然无法终止进程，Sanlock 将会依赖 *watchdog* 守护进程来重启这个主机。

每次当主机的 VDSM 更新它的 *hostid* 并在 *lockspace* 中写时间戳时，*watchdog* 守护进程都会收到一个 *pet*。当 VDSM 不能进行这些操作时，*watchdog* 守护进程将无法收到 *pet*。如果 *watchdog* 守护进程在一定时间内仍然没有收到 *pet*，它将会重启主机。这将保证 SPM 资源可以被释放，从而可以被其它主机使用。

## 2.11. 精简分配 (Thin Provisioning) 和存储过度分配 (Over-Commitment)



Red Hat Virtualization Manager 提供了存储分配策略来优化虚拟环境中的存储使用。使用精简分配（thin provisioning）策略可以根据虚拟环境中的实际使用情况实现存储资源的“过度分配（over-commit）”功能。

“存储过度分配”是指分配给虚拟机的存储总量比存储池中所具有的物理存储总量要大。通常情况下，虚拟机不会使用分配给它们的全部存储资源。从用户的角度来看，使用精简分配功能的虚拟机完全具有了所有定义的存储空间；而实际上，只有一部分存储空间被实际分配给虚拟机。



### 注意

虽然 Red Hat Virtualization Manager 提供了它自己的精简分配功能，但是如果后台存储设备本身具有精简分配的功能，您应该使用存储设备本身的功能。

存储过度分配需要定义一个阈值。VDSM 会比较逻辑存储和实际的存储使用情况，它通过这个定义的阈值来保证需要写到磁盘中的数据小于实际保存它的逻辑卷。QEMU 在一个逻辑卷中指定写操作可以使用的最高的偏移值，这个值就是写操作在存储中可以被使用的最高点。VDSM 会监测这个最高的偏移值来保证存储的实际使用不会超过预先定义的阈值。只要 VDSM 认为最高的偏移值低于这个阈值，Red Hat Virtualization Manager 就可以认为有足够的存储来保证系统可以正常运行。

当 QEMU 所指定的使用最高偏移值超过了阈值时，VDSM 就会通知 Red Hat Virtualization Manager 磁盘镜像很快会超过它的逻辑卷的容量，Manager 则会请求 SPM 主机来扩展逻辑卷。只要数据中心中的存储域有足够的空间，这个扩展逻辑卷的过程就可以继续进行。如果存储域中已经没有空闲空间时，您需要手工添加更多的存储设备来增加存储容量。

## 2.12. 逻辑卷扩展

Red Hat Virtualization Manager 使用精简分配策略来为一个存储池实现存储过度分配功能。虚拟机的操作会产生数据，使用精简分配磁盘镜像的虚拟机最终将会出现所写的的数据大于它所在的逻辑卷的情况。当这个情况发生时，逻辑卷扩展控制功能会为虚拟机的正常运行提供更多的存储。

Red Hat Virtualization 提供了一个 LVM 的精简分配机制。当使用 QCOW2 格式的存储时，Red Hat Virtualization 使用主机的系统进程 *qemu-kvm* 来为磁盘上的存储块和逻辑块之间建立一个映射关系，这可以实现现在小逻辑卷上创建大逻辑磁盘的功能，例如，可以在 1GB 的逻辑卷上创建一个 100GB 的逻辑磁盘。当 *qemu-kvm* 超过了 VDSM 所设置的阈值时，本地的 VDSM 会向 SPM 发出一个为逻辑卷增加 1GB 空间的请求。运行需要扩展逻辑空间的虚拟机的主机上的 VDSM 会通知 SPM VDSM 需要更多的存储空间。SPM 将扩展逻辑卷，SPM VDSM 通知主机的 VDSM 来更新逻辑组信息，并完成扩展的操作。

逻辑卷扩展的操作并不需要主机知道哪个主机是 SPM，即使需要扩展逻辑卷的主机本身就是 SPM 也可以，所有关于扩展的交流信息都是通过一个“存储邮箱”进行的。存储邮箱被保存在数据存储域中的一个专门的逻辑卷中。当一个主机需要 SPM 扩展逻辑卷时，它会在存储邮箱的相应区域中留下一条信息，而 SPM 会定期查看邮箱中的信息，执行逻辑卷扩展操作，并向发出扩展请求的主机返回一条信息。当主机发出扩展请求后，它会每隔 2 秒来检查它的新邮件。如果成功接收到了扩展请求的返回邮件，主机就会在设备映射表中刷新逻辑卷的信息，从而可以使用新分配的存储。

当一个存储池中的物理存储设备没有可用空间时，QEMU 会返回一个 **enospc error**。如果出现这个错误，正在运行的虚拟机会被自动暂停，这时需要手工来为卷组添加新的 LUN。

当一个新的 LUN 被添加到卷组中时，SPM 会自动把新增的存储分配给需要的逻辑卷，从而使相关的虚拟机可以自动被恢复运行。

## 第 3 章 网络

### 3.1. 网络架构

Red Hat Virtualization 的网络可以从 3 个方面介绍：基本网络、集群中的网络和主机网络配置。基本网络包括了实现网络功能的基本软件和硬件；集群中的网络包括了在集群中的资源间的网络连接（如主机、逻辑网络和虚拟机）；主机网络配置包括在一个主机上被支持的网络配置。

一个有良好设计的网络将会为用户提供一个高性能的网络，并可以顺利实现虚拟机的迁移。而一个没有经过良好设计的网络可能为系统的使用和维护带来许多问题（如网络响应速度太慢、虚拟机迁移和克隆失败等）。

### 3.2. 介绍：基本网络元素

Red Hat Virtualization 使用以下元素来为虚拟机、主机和虚拟环境提供网络服务：

- ✧ 网卡（NIC - Network Interface Controller）
- ✧ 网桥（bridge）
- ✧ 网络绑定（bond）
- ✧ 虚拟网卡（VNIC）
- ✧ 虚拟局域网（VLAN）

网卡（NIC）、网桥和虚拟网卡提供了主机、虚拟机、局域网和 Internet 间的网络功能。网络绑定（bond）和虚拟局域网（VLAN）可以增强网络安全性、提供网络容错功能、增加网络性能。

### 3.3. 网络接口控制器

*NIC（Network Interface Controller - 网络接口控制器）* 通常被称为网卡，它是一个网络或 LAN 适配器，用来把计算机和计算网络进行连接。NIC 运行在网络层和数据连接层来为所在的机器提供网络连接功能。在 Red Hat Virtualization 环境中的主机都最少需要有一个 NIC，但通常情况下主机都会有多个网卡。

一个物理网卡可以有多个虚拟网卡（VNIC）和它相连，而一个虚拟网卡可以看做为一个虚拟机的物理网卡。为了区分虚拟网卡和物理网卡，Red Hat Virtualization Manager 会为每个虚拟网卡分配一个独立的 MAC 地址。

### 3.4. 网桥

*网桥（bridge）* 是在数据包交换网络中使用数据包转发的软件设备。通过使用网桥，多个网络接口设备可以共享同一个物理网卡，而每个网络接口设备都会以独立的物理设备的形式出现在网络中。网桥会检查一个数据包的源地址来决定相关的目标地址，一旦获得了目标地址的信息，它会在一个表中添加这个地址以供以后使用。通过使用网桥，主机可以把网络数据重新定向到使用虚拟网卡的、相应的虚拟机上。

在 Red Hat Virtualization 中，逻辑网络是通过网桥来实现的。一个 IP 地址会分配给网桥而不是主机本身的物理网络接口，这个 IP 地址不需要和连接到这个网桥上的虚拟机的网络处于同一个子网中。如果分配给网桥的 IP 地址和虚拟机处于同一个子网中，网桥所在的主机将可以被虚拟机直接访问，我们通常不推荐在虚拟主机上运行可以被访问的网络服务。虚拟机通过它们的虚拟网卡连接到逻辑网络中，虚拟机上的每个虚拟网卡都可以通过 DHCP 或静态分配来获得独立的 IP 地址。网桥可以连接到主机以外的系统，但这不是必须的。

网桥和以太网连接都可以设置自定义属性。VDSM 会把网络定义和自定义属性传递给设置网络的 hook 脚本。

## 3.3. 绑定

**绑定 (bond)** 是由多个网卡组合成的一个单一的、由软件定义的网络设备。因为一个绑定是由多个网卡组成的，因此它可以提供比单一网卡更高的网络传输速度，并提供了更好的网络容错功能（绑定只有在所有的网卡都出现问题时才会停止工作）。但是，绑定设备有一个限制：绑定必须由相同型号的网卡组成。

绑定设备的数据包传输算法是由绑定的模式所决定的。



### 重要

模式 1、2、3 和 4 支持虚拟机网络（使用网桥）和非虚拟机网络（无网桥）；模式 0、5 和 6 只支持非虚拟机网络（无网桥）。

## 绑定模式

Red Hat Virtualization 使用 Mode 4 作为默认的模式，它同时也支持以下绑定模式：

### 模式 0 (*round-robin policy*)

传输的数据包会顺序使用网卡。它会首先使用绑定中的第一个有效的网卡，最后使用最后一个网卡。模式 0 提供了网络容错和网络负载均衡的功能，但它不能和网桥一起使用，因此与虚拟机逻辑网络不兼容。

### 模式 1 (*active-backup policy*)

绑定中的一个网络接口被设置为活跃接口来处理网络数据，其它网络接口都为备份接口。如果活跃接口出现了问题，备份接口中的一个网络接口会成为活跃接口来继续处理网络数据。使用模式 1 的绑定设备的 MAC 地址只在一个端口上可见，这可以避免因为切换活跃接口所造成的 MAC 地址改变所带来的混淆。模式 1 提供了网络容错的功能。

### 模式 2 (*XOR policy*)

模式 2 (XOR policy) 会对源和目标 MAC 地址进行 XOR 操作，所获得的结果再对“次要网卡”的数量进行取模。系统会根据最后所获得的结果来选择用来传输数据包接口。它保证了对于每个目标 MAC 地址，相同的接口都会被选择。模式 2 提供了容错和负载均衡的功能。

### 模式 3 (*broadcast policy*)

使用绑定中的所有网卡来传输数据包。它提供了网络容错的功能。

### 模式 4 (*IEEE 802.3ad policy*)

模式 4 (IEEE 802.3ad policy) 会创建一个整合的组，这个组会共享网速和网络双工 (duplex) 设置。模式 4 会根据 IEEE 802.3ad 标准使用活动组中的所有网络接口。

### 模式 5 (*adaptive transmit load balancing policy*)

模式 5 保证所有出站的网络流量会根据每个接口的负载进行分配，而所有入站的网络流量都被当前的接口所接收。如果用来接收网络流量的接口出现故障，另外一个网络接口会被指定来接收网络流量。因为模式 5 不能和网桥一起使用，所以它与虚拟机网络不兼容。

### 模式 6 (*adaptive load balancing policy*)

Mode 5 的功能再加上不需要特殊的网络交换要求的 IPv4 网络数据接收负载均衡功能。它在处理接收负载时使用 ARP。因为模式 6 不能与网桥一起使用，所以它与虚拟机逻辑网络不兼容。

### 3.6. 绑定的交换配置

绑定的交换配置会因为硬件的不同而有所不同。请参阅您的操作系统中关于实施和配置网络的信息。



#### 重要

对于每一类交换，用户在设置交换绑定时需要使用 *Link Aggregation Control Protocol* (LACP) 协议而不要使用 *Cisco Port Aggregation Protocol* (PAgP) 协议。

### 3.7. 虚拟网络接口卡（虚拟网卡）

虚拟网络接口卡是基于主机的物理网卡的虚拟网络接口。每一个主机可以有多个物理网卡，而每个物理网卡可以有多个虚拟机网络接口卡（虚拟网卡）。

当您为虚拟机添加一个虚拟网卡时，Red Hat Virtualization Manager 会在虚拟机、虚拟网卡本身和虚拟网卡所基于的主机的物理网卡间创建一定的关联。虚拟网卡所基于的主机的物理网卡上会创建一个新的虚拟网卡和 MAC 地址。当虚拟机第一次启动时，**libvirt** 会为虚拟网卡分配一个 PCI 地址。这样，虚拟机就可以使用 MAC 地址和 PCI 地址（如 **eth0**）指定虚拟网卡。

如果虚拟机是通过模板或快照创建的，分配 MAC 地址以及把这些 MAC 地址和 PCI 地址相关联的过程会有所不同。当模板或快照中已经包括了 PCI 地址时，通过这个模板或快照所创建的虚拟机上的虚拟网卡的顺序会根据这些 PCI 地址和 MAC 地址被分配；如果模块中没有包括 PCI 地址，基于这个模板所创建的虚拟机上的虚拟网卡会根据虚拟机网卡的名称被创建；如果快照中没有包括 PCI 地址，Red Hat Virtualization Manager 会根据快照来为虚拟网卡分配新的 MAC 地址。

创建后，虚拟网卡会被添加到网桥设备中。网桥将用来处理虚拟机和虚拟机网络的连接。

在一台主机上运行 **ip addr show** 命令会显示这个主机上的与虚拟机相关的所有虚拟网卡信息。另外，它还会显示虚拟网络所使用的网桥信息，以及主机上的所有网卡信息。

```
[root@rhev-host-01 ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
    link/ether 00:21:86:a2:85:cd brd ff:ff:ff:ff:ff:ff
    inet6 fe80::221:86ff:fea2:85cd/64 scope link
        valid_lft forever preferred_lft forever
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN
qlen 1000
    link/ether 00:21:6b:cc:14:6c brd ff:ff:ff:ff:ff:ff
5: vdsmdummy: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN
    link/ether 4a:d5:52:c2:7f:4b brd ff:ff:ff:ff:ff:ff
6: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
7: bond4: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
8: bond1: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
```



```

9: bond2: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
10: bond3: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
11: ovirtmgmt: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
    state UNKNOWN
    link/ether 00:21:86:a2:85:cd brd ff:ff:ff:ff:ff:ff
    inet 10.64.32.134/23 brd 10.64.33.255 scope global ovirtmgmt
    inet6 fe80::221:86ff:fea2:85cd/64 scope link
        valid_lft forever preferred_lft forever

```

这个命令显示了以下设备：一个环回设备（**lo**）、一个以太网设备（**eth0**）、一个无线设备（**wlan0**）、一个 VDSM 虚拟设备（**;vdsmdummy;**）、5 个绑定设备（**bond0**、**bond4**、**bond1**、**bond2**、**bond3**）和一个网桥（**ovirtmgmt**）。

上面所显示的虚拟网络都属于同一个网桥设备和逻辑网络。您可以使用 **brctl show** 命令来显示网桥都包括哪些虚拟网卡：

```

[root@rhev-host-01 ~]# brctl show
bridge name bridge id STP enabled interfaces
ovirtmgmt 8000.e41f13b7fdd4 no vnet002
        vnet001
        vnet000
        eth0

```

这个 **brctl show** 命令输出显示了 virtio 虚拟网卡包括在 **ovirtmgmt** 网桥中。和这个虚拟网卡所关联的所有虚拟机都属于 **ovirtmgmt** 逻辑网络。**eth0** 网络接口也是 **ovirtmgmt** 网桥的一部分。**eth0** 设备和交换机连接来提供到主机以外的网络连接。

### 3.8. 虚拟局域网（VLAN）

VLAN（*虚拟局域网*）是一个可以在网络数据包中使用的属性，网络数据包可以被“标识”为属于不同的 VLAN。VLAN 提供了一个在数据交换层分离网络数据的安全功能，不同的 VLAN 完全独立。Red Hat Virtualization Manager 支持 VLAN 功能，并可以使用它来标识和重定向 VLAN 数据，但是 VLAN 的实现还需要所使用的网络交换机支持 VLAN 功能。

在网络交换机一级，每个端口都会被分配一个 VLAN 标记。交换机会为从特定端口发出的数据添加一个 VLAN 标记，并确保相应的响应数据带有相同的 VLAN 标记。一个 VLAN 可以跨越多个交换机，带有 VLAN 标识的网络数据只能被连接到特定交换机端口的、带有正确 VLAN 的机器所访问。一个端口可以被标识为带有多个 VLAN，这可以使不同 VLAN 网络中的数据被发送到这个端口上。然后，使用所在机器上的软件来对接收到的数据进行处理。

### 3.9. 网络标签（Network Label）

使用网络标签（Network Label），可以大大简化一些逻辑网络管理的工作。

网络标签就是和一个逻辑网络或主机的物理网络接口相关联的一组文字。网络标签的长度没有限制，但它只能包括大小写字母、下划线和分号。空格和其它特殊符合不被支持。

为逻辑网络或主机的物理网络接口加一个网络标签后，它们就可以和有相同网络标签的逻辑网络或主机的物理网络接口以下面的形式相关联：

#### 网络标签关联

- ✧ 当您为一个逻辑网络添加了一个网络标签后，这个逻辑网络将会被自动和有相同网络标签的主机物理网络接口相关联。
- ✧ 当您为一个主机的物理网络接口添加了一个网络标签后，具有这个网络标签的所有逻辑网络都会和这个主机的物理网络接口相关联。
- ✧ 修改已经被附加到一个逻辑网络或主机的物理网络接口的网络标签等同于：先删除了网络标签，然后再添加了一个新网络标签。附加了这个网络标签的逻辑网络和主机的物理网络接口之间的关联也会被更新。

### 网络标签和集群

- ✧ 当一个有网络标签的逻辑网络被添加到一个集群中，它会被自动添加到在这个集群中的具有相同网络标签的主机物理网络接口上。
- ✧ 当一个有网络标识的逻辑网被从一个集群中删除后，它会被自动取消和在这个集群中的具有相同网络标签的主机物理网络接口的关联。

### 网络标签和有用户角色的逻辑网络

- ✧ 当一个有网络标识的逻辑网络被设为“显示网络”或“迁移网络”时，它会在主机物理网络接口中被配置为通过 DHCP 获得一个 IP 地址。

## 3.10. 集群网络

集群层的网络项包括：

- ✧ 集群
- ✧ 逻辑网络

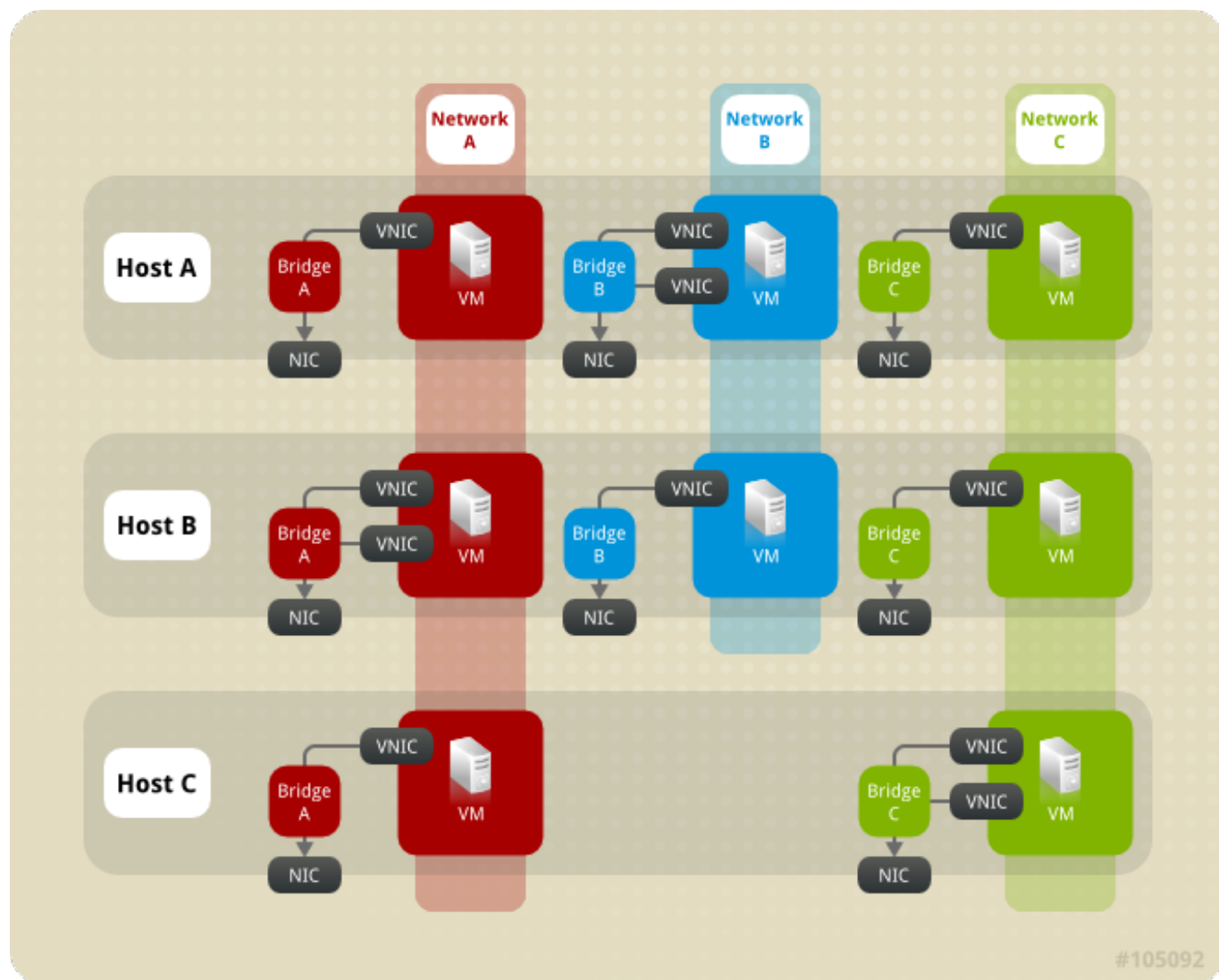


图 3.1. 一个集群内的网络

一个数据中心就是由多个集群组成的一个逻辑组，而一个集群就是由多个主机组成的一个逻辑组。[图 3.1 “一个集群内的网络”](#) 展示了一个集群中所包括的项。

在一个集群中的所有主机都可以访问相同的存储域，同时它们也在集群这一层来关联逻辑网络。对于一个虚拟机逻辑网络来说，为了使虚拟机可以使用它，它必须通过 Red Hat Virtualization Manager 在集群中的所有主机上定义并配置；而对于其它类型的逻辑网络，它们只需要在使用这些网络的主机上定义。

多主机网络配置的功能在数据中心中被支持。当一个网络设置更新后，这个网络配置更新会自动应用到这个数据中心中所有分配了这个网络的主机上。

### 3.11. 逻辑网络

Red Hat Virtualization 环境使用逻辑网络来根据网络数据类型对不同网络进行分离。例如，在安装 Red Hat Virtualization 时默认创建的 `ovirtmgmt` 网络被用来作为处理 Manager 和主机间的管理通信所需的网络。一般情况下，具有类似要求和使用情况的多个网络通信可以组成一个逻辑网络。系统管理员通常会创建一个存储网络和一个显示网络来把这两种网络流量分离来，从而达到提高系统性能，方便故障排除的效果。

逻辑网络的类型包括：

- ✧ 用来处理虚拟机网络流量的逻辑网络，
- ✧ 不处理虚拟机网络流量的逻辑网络，

- ✧ 可选的逻辑网络，
- ✧ 必需的逻辑网络。

所有逻辑网络都可以是“必需的”逻辑网络或“可选的”逻辑网络。

逻辑网络在数据中心一级被定义，并被添加到主机上。为了使一个“必需的”逻辑网络可以正常工作，它需要被添加到相应集群中的所有主机上。

Red Hat Virtualization 环境中的每个虚拟机逻辑网络都必须由一个主机上的一个网桥设备支持。当为一个集群定义了一个新的虚拟机逻辑网络后，在这个集群的所有主机上都需要创建一个匹配的网桥设备，这样，这个新的虚拟机逻辑网络才可以被虚拟机使用。Red Hat Virtualization Manager 会自动为虚拟机逻辑网络创建所需要的网桥设备。

Red Hat Virtualization Manager 为虚拟机逻辑网络所创建的网桥设备需要和主机上的一个网络接口相关联。如果和网桥相关的主机网络接口已经被其它网络所使用，则新加入的网络接口将同样可以共享那些已经连接到主机网络接口中的网络。当虚拟机被创建并被添加到一个特定的逻辑网中时，它们的虚拟网卡会被添加到那个逻辑网所在的网桥中。这样，虚拟机就可以和连接到相同网桥中的其它设备进行网络交流。

不处理虚拟机网络流量的逻辑网络会和主机的网卡直接进行关联。

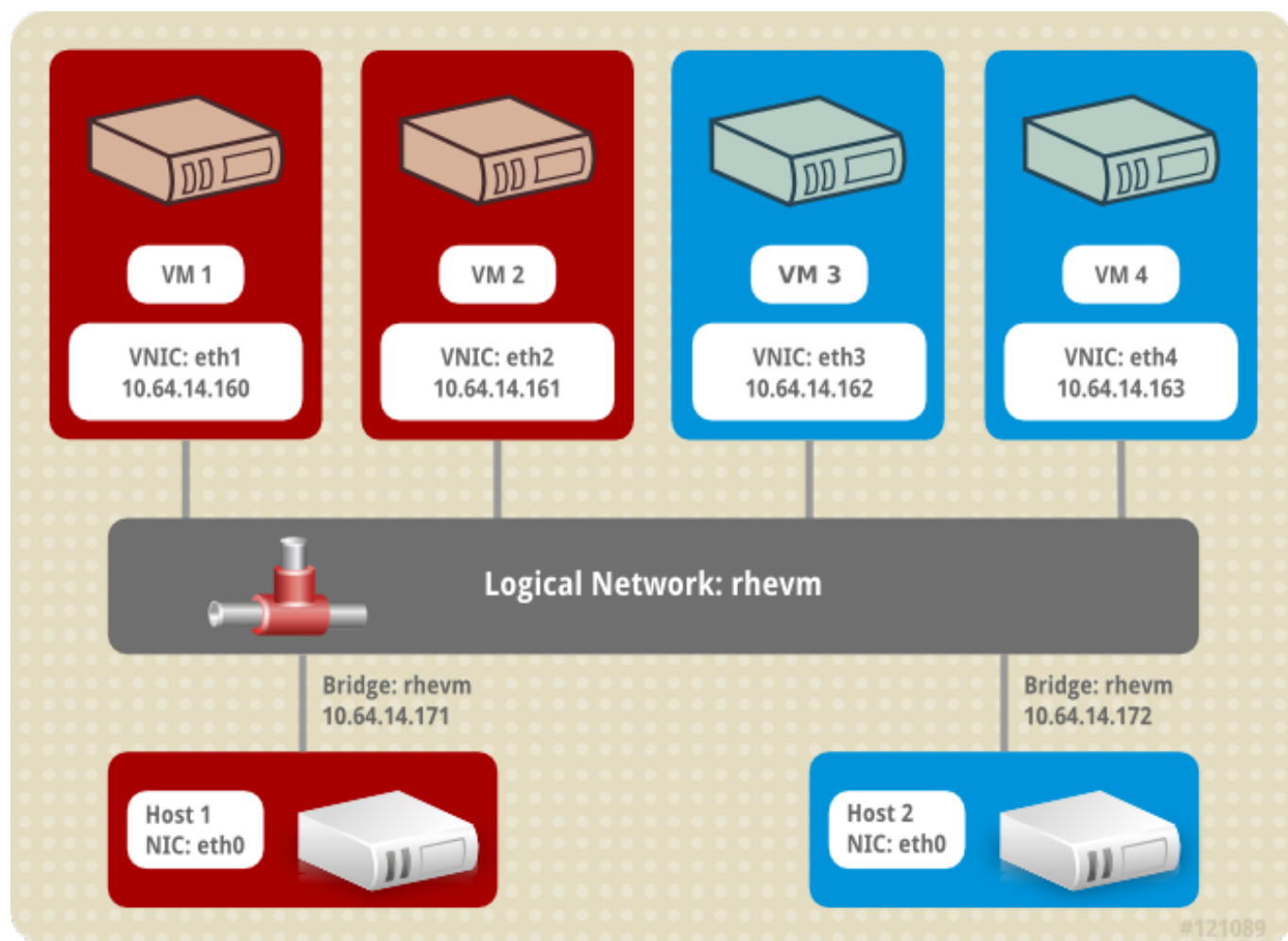


图 3.2. ovirtmgmt 逻辑网络。

### 例 3.1. 逻辑网络使用示例。

名为“Purple”的数据中心中的名为“Pink”的集群中有两个主机：“Red”和“White”。“Red”和“White”都使用默认的逻辑网络 `ovirtmgmt`。数据中心“Purple”的系统管理员需要把与测试一个 web 服务器相关的网络流量和其它网络隔离开，因此系统管理员决定把这个 web 服务器和一些虚拟机添加到一个名为 `network_testing` 的新逻辑网络中。

首先，系统管理员在数据中心“Purple”中定义了一个逻辑网络，然后把它加入到集群“Pink”中。因为只能在主机处于维护模式时才可以部署逻辑网络，所以管理员把所有运行的虚拟机都迁移到“Red”上，并把“White”设为维护模式。然后，系统管理员需要编辑与包括在需要添加到网桥中的物理网卡相关联的网络，相应的网络接口的连接状态会从 **Down** 变为 **Non-Operational**（因为相关的网桥还没有在集群中的所有主机上被设置，所以现在的“连接状态”是“Non-Operational”）。现在，管理员需要把主机的物理网络接口添加到 `network_testing` 网络中。操作完成后，管理员需要激活“White”，然后把“Red”上运行的虚拟机迁移到“White”上，并在“Red”上重复在“White”上进行的操作。

当把逻辑网络 `network_testing` 与主机“Red”和主机“White”上的物理网络接口相关联后，逻辑网络 `network_testing` 就变为 **Operational**，并可以开始被虚拟机使用。

### 3.12. 必需的网络、可选网络和虚拟机网络

一个“必需的”网络就是一个逻辑网络，它需要对一个集群中的所有主机都有效。当一个主机的“必需的”网络无法工作时，在它上面运行的虚拟机会被迁移到另外一个主机上，而具体的迁移过程取决于所选择的调度策略。这一点对于运行关键任务服务的虚拟机非常重要。

一个可选网络就是一个没有被声明为**必需的**的逻辑网络。可选网络可以只在需要它的主机上有效。有或没有可选的网络不会影响到一个主机的 **Operational** 状态。当一个不是“必需的”网络无法工作时，使用它的虚拟机不会被迁移到另一个主机上。这可以避免因为虚拟机迁移而产生的、不必要的 I/O 负载。请注意，当一个逻辑网络被创建并被加入到集群后，**必需的**选项会被默认选择。

要改变网络的 **Required** 设置，使用管理门户，选一个网络，点**集群**标签页，点**管理网络**按钮。

虚拟机网络（**VM network**）是那些只处理虚拟机网络流量的逻辑网络。虚拟机网络可以是必需的网络，也可以是可选网络。使用一个可选虚拟机网络的虚拟机只会在带有这个网络的主机上启动。

### 3.13. 虚拟机连接

在 Red Hat Virtualization 中，当一个虚拟机被创建时，它的网卡会被添加到一个逻辑网络中。这样，它就可以和同一个网络中的其它对象进行网络交流。

从主机的角度来看，当虚拟机被添加到一个逻辑网络时，虚拟机网卡所基于的虚拟网卡（vNIC）会被添加到逻辑网络所使用的网桥设备中。例如，一个连接到 `ovirtmgmt` 逻辑网络的虚拟机，它的虚拟网卡会被添加到运行这个虚拟机的主机的 `ovirtmgmt` 网桥设备中。

### 3.14. 端口镜像

端口镜像（port mirroring）会把指定逻辑网络和主机上的第 3 层网络流量复制到一个虚拟机的虚拟网络接口上。这样，通过这个虚拟机就可以进行网络纠错、网络优化、网络入侵检测以及对在同一个主机和逻辑网络中运行的虚拟机进行监控。

端口镜像只复制一个主机和一个逻辑网间内部的网络数据，它不会增加这个主机以外的网络流量。但是，启用了端口镜像功能的主机会比其它主机消耗更多 CPU 和内存资源。

端口镜像可以通过逻辑网络的 vNIC 配置集被启用或禁用，它具有以下的限制：

- ✱ 不支持对在配置集中启用了端口镜像功能的 vNIC 进行“热插”。



- ✧ 当 vNIC 配置集被附加到一个虚拟机时，端口镜像不能被改变。

鉴于以上限制，我们推荐在一个额外的专用 vNIC 配置集中启用端口镜像。



## 重要

启用端口镜像会降低其它网络用户的隐私保护。

## 3.15. 主机网络配置

主机上包括的常见网络配置类型：

- ✧ 网桥和网卡配置。
- ✧ 网桥、VLAN（虚拟局域网）和网卡配置。
- ✧ 网桥、网络绑定和 VLAN 配置。
- ✧ 多网桥、多 VLAN 和网卡配置。

## 3.16. 网桥配置

Red Hat Virtualization 中最简单的配置就是网桥 + 网卡的配置。如 [图 3.3 “网桥和网卡配置”](#) 所示，这个配置使用一个网络来把一个或多个虚拟机和主机的网络进行连接。

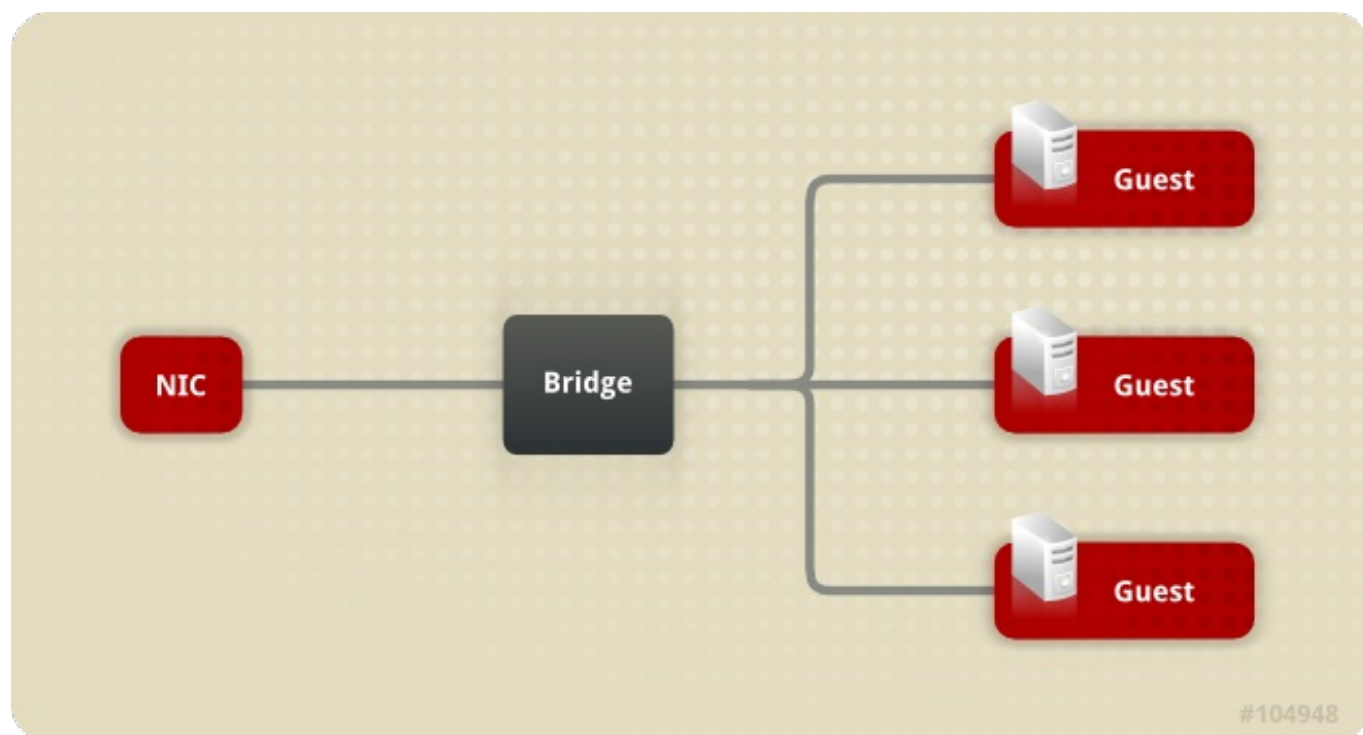


图 3.3. 网桥和网卡配置

这种配置的一个实例就是安装 Red Hat Virtualization Manager 时自动创建的 **ovirtmgmt** 网桥。在安装的过程中，Red Hat Virtualization Manager 在主机上安装 **VDSM**，**VDSM** 的安装过程包括创建 **ovirtmgmt** 网桥。**ovirtmgmt** 网桥然后可以获得主机的 IP 地址来实现主机的管理网络功能。

### 3.17. VLAN 配置

图 3.4 “网桥、VLAN 和网卡配置”展示了另外一种配置，它包括了一个连接到主机网卡和网桥的虚拟 LAN (VLAN)。

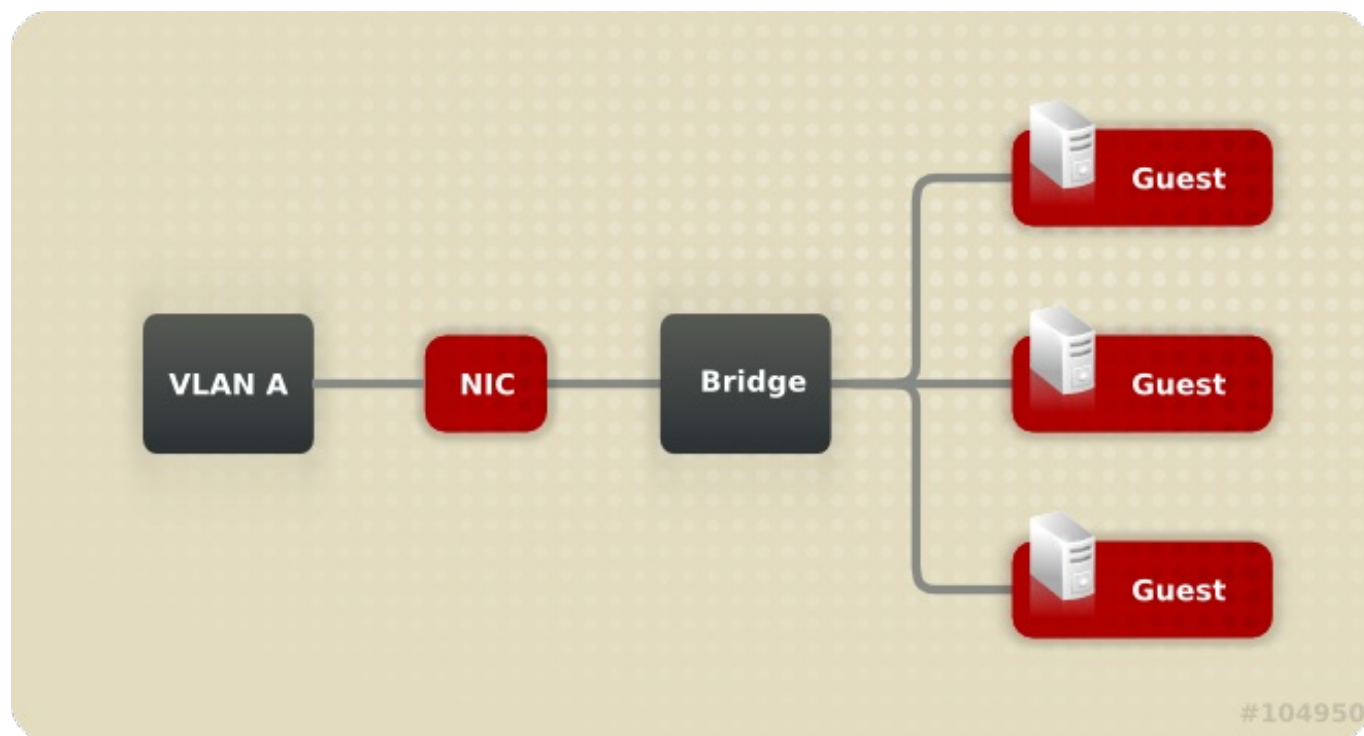


图 3.4. 网桥、VLAN 和网卡配置

使用一个 VLAN 为这个网络中的数据传输提供了一个安全的通道。另外，使用多个 VLAN 还可以实现把多个网桥连接到一个网卡的功能。

### 3.18. 网桥和绑定配置

图 3.5 “网桥、绑定和网络配置”所显示的配置中包括一个网络绑定，多个主机网卡通过绑定和同一个网桥和网络进行连接。

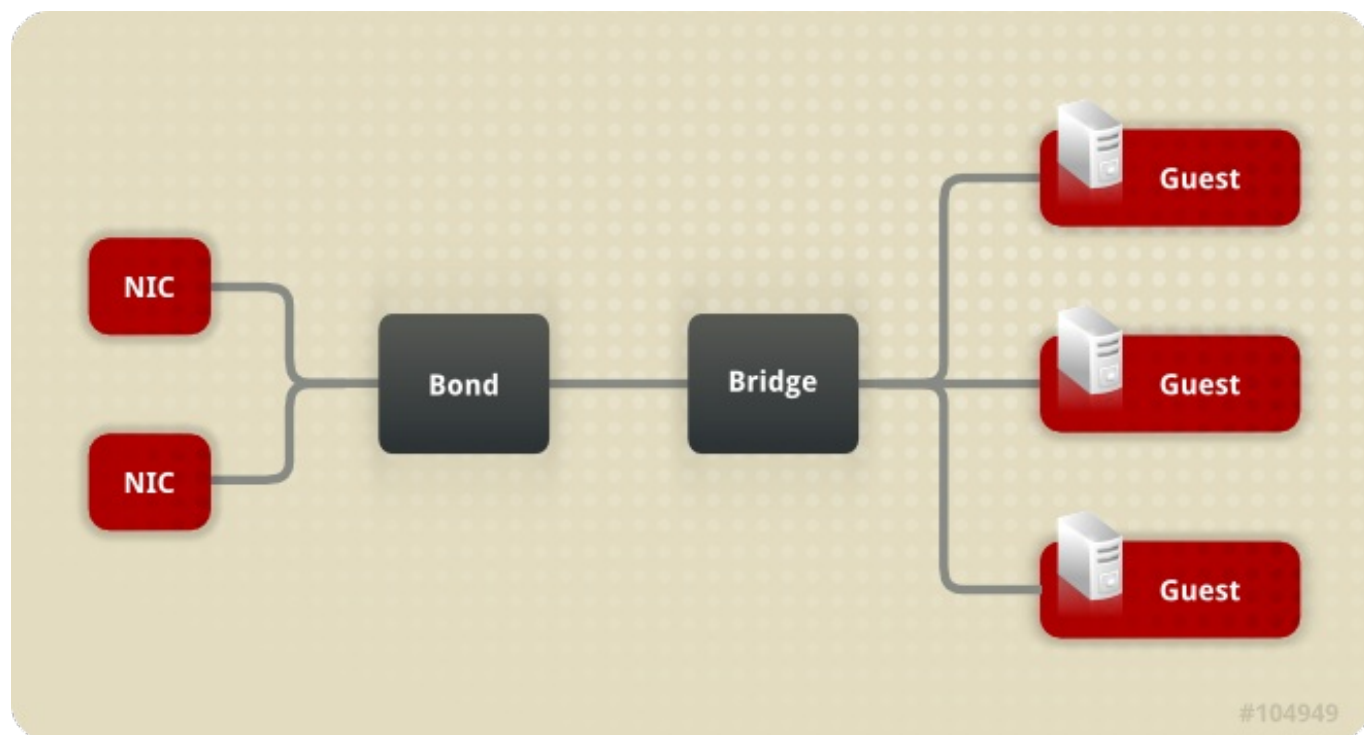


图 3.5. 网桥、绑定和网络配置

通过其中的绑定创建了一个逻辑连接来把两个（或更多）物理以太网连接起来。根据所选择使用的绑定模式，这种配置可以提供网卡容错、扩展网络带宽等好处。

### 3.19. 多网桥、多 VLAN 和单网卡配置

图 3.6 “多网桥、多 VLAN 和网卡配置”所显示的配置把一个网卡连接到两个 VLAN（这同时需要网络交换机的支持）。主机使用两个独立的 VNIC 来分别处理两个 VLAN 网络数据，而每个 VLAN 会连接到不同的网桥中。同时，每个网桥可以被多个虚拟机使用。

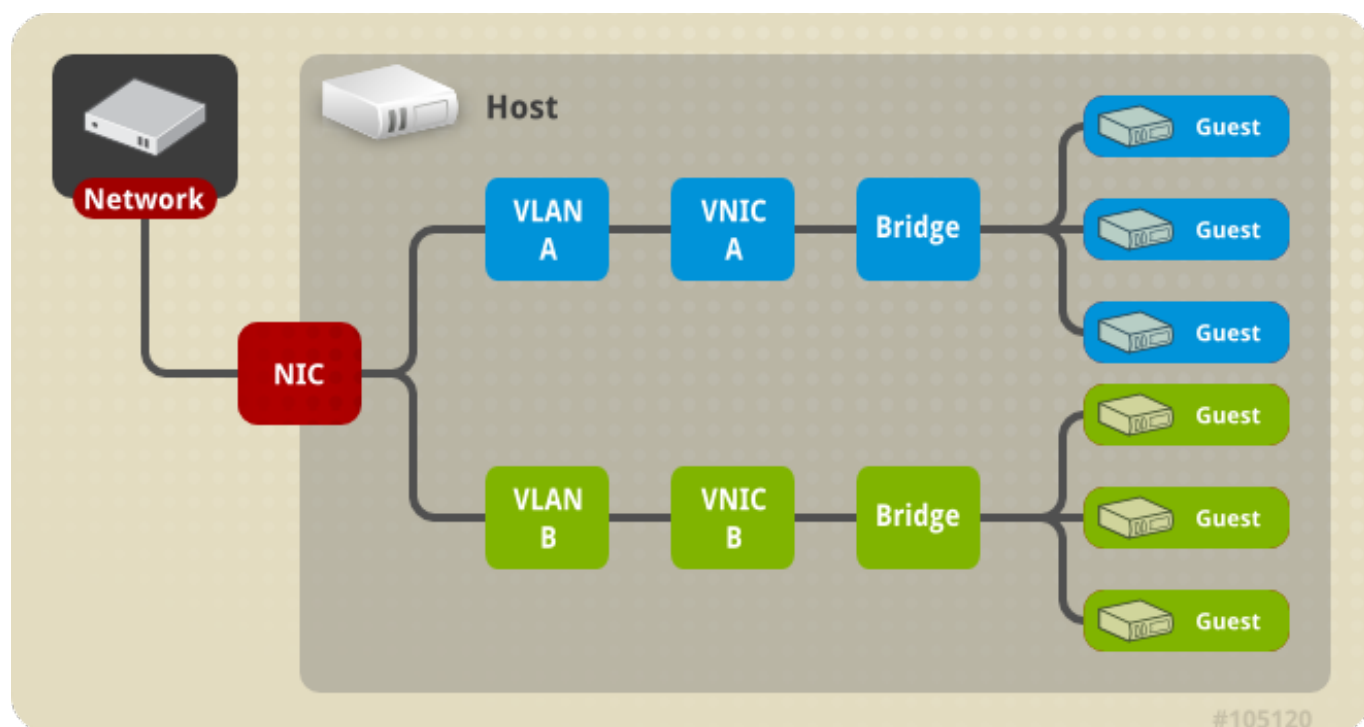


图 3.6. 多网桥、多 VLAN 和网卡配置



### 3.20. 多网桥、多 VLAN 和单绑定配置

图 3.7 “多网桥、多 VLAN 和由多个网卡组成的绑定连接” 显示的配置使用多个网卡组成一个绑定设备来连接到多个 VLAN。

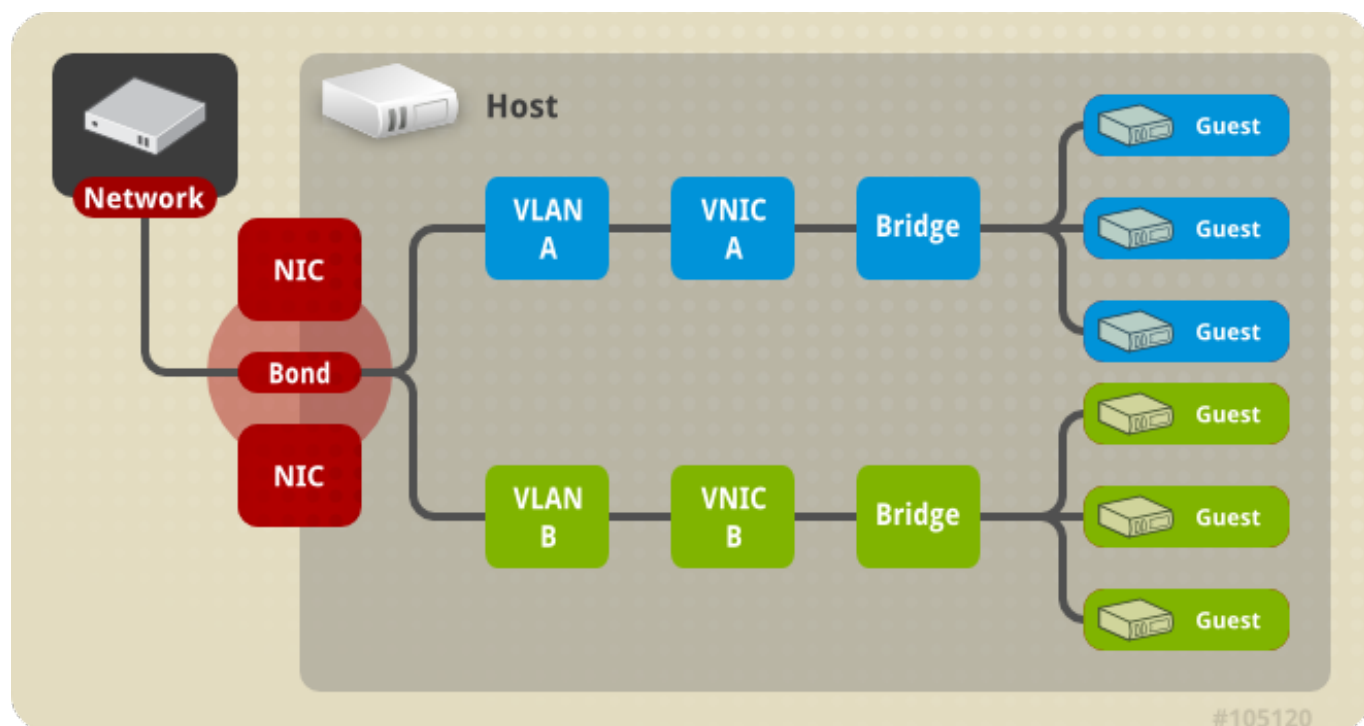


图 3.7. 多网桥、多 VLAN 和由多个网卡组成的绑定连接

这个配置中的 VLAN 通过绑定和网卡进行连接。每个 VLAN 连接到不同的网桥，每个网桥可以连接一个或多个虚拟机。

## 第 4 章 电源管理

### 4.1. 电源管理和隔离介绍

当设置了电源管理和隔离（fence）功能后，Red Hat Virtualization 环境将会提供更好的灵活性和稳定性。电源管理功能将可以使 Red Hat Virtualization Manager 控制主机的电源操作，其中最重要的一点是可以在主机出现故障时重新启动主机。隔离功能被用来把出现故障的主机从 Red Hat Virtualization 环境中分离，从而使整个虚拟环境正常运行。当被隔离的主机的故障被排除后，它可以重新返回正常工作的状态，并被重新加入到原来的虚拟环境中。

电源管理和隔离功能需要使用独立于操作系统的专用硬件来实现。Red Hat Virtualization Manager 使用电源管理设备的 IP 地址或主机名来访问它。在 Red Hat Virtualization 中，电源管理设备和隔离设备是相同的。

### 4.2. 在 Red Hat Virtualization 环境中使用代理进行电源管理

Red Hat Virtualization Manager 不直接和隔离设备进行通讯，它使用一个代理来向主机的电源管理设备发送电源管理命令。Manager 需要使用 VDSM 来进行电源管理设备的操作，因此环境中还需要另外一个主机作为隔离代理。

您可以选择：

- ✧ 需要隔离功能的主机所在的同一个集群中的任何主机。
- ✧ 需要隔离功能的主机所在的同一个数据中心中的任何主机。

隔离代理主机的状态有两种：*UP* 和 *Maintenance*。

### 4.3. 电源管理

Red Hat Virtualization Manager 可以重启那些处于“无法正常工作（non-operational）”或“无响应（non-responsive）”状态的主机；或为省电关闭那些有低利用率的主机，但这些操作需要电源管理设备被正确配置。Red Hat Virtualization 环境支持以下电源管理设备：

- ✧ *American Power Conversion*（apc）。
- ✧ *Bladecenter*。
- ✧ *Cisco Unified Computing System*（cisco\_ucs）。
- ✧ *Dell Remote Access Card 5*（drac5）。
- ✧ *Dell Remote Access Card 7*（drac7）。
- ✧ *Electronic Power Switch*（eps）。
- ✧ *HP BladeSystem*（hpblade）。
- ✧ *Integrated Lights Out*（ilo、ilo2、ilo3、ilo4）。
- ✧ *Intelligent Platform Management Interface*（ipmilan）。
- ✧ *Remote Supervisor Adapter*（rsa）。
- ✧ *rsb*。
- ✧ *Western Telematic, Inc*（wti）。



## 注意

**apc** 隔离代理不支持 APC 5.x 电源管理设备，您需要使用 **apc\_snmp** 隔离代理。

Red Hat Virtualization Manager 使用 *隔离代理 (fence agent)* 来和电源管理设备进行交流，系统管理员可以使用 Red Hat Virtualization Manager 配置电源管理的隔离代理（使用电源管理设备支持的参数）。简单的配置（被所有电源管理设备所支持的操作）可以通过系统提供的图形用户界面进行，而特殊的配置选项（只适用于特定隔离设备的选项）也可以通过这个界面被输入，但系统不会对它们进行任何处理，而直接把这些配置选项传递给隔离设备。被所有电源管理设备支持的操作配置包括：

- ✦ **Status**：检查主机的状态。
- ✦ **Start**：启动主机。
- ✦ **Stop**：关闭主机。
- ✦ **Restart**：重启主机（实际上是通过执行 stop、wait、status、start、wait、status 操作实现的）。

作为一个最佳的实践规则，您需要在初始配置完成后马上测试电源管理功能，并定期对运行环境中的电源管理功能进行测试。

一个稳定的系统需要在环境中的所有主机上正确地配置电源管理设备。在出现问题的主机上，Red Hat Virtualization Manager 可以使用隔离代理跳过操作系统来直接和主机上的电源管理设备直接进行交流，并通过重启主机来把有问题的主机从虚拟环境中隔离。如果被隔离的主机上有运行的高可用性虚拟机，Manager 可以安全地把这些高可用性虚拟机迁移到其它主机上；如果被隔离的主机是 SPM，Manager 可以把 SPM 角色分配给其它主机。

## 4.4. 隔离

在 Red Hat Virtualization 环境中，隔离操作（fencing）就是由 Manager 通过使用隔离代理发起的、由电源管理设备负责执行的主机重启操作。隔离操作可以使集群对意料外的主机故障做出相应的响应；或根据预先设定的规则实现省电、负载均衡、虚拟机可用性策略等功能。

隔离功能可以保证 SPM 角色一直被一个正常工作的主机所具有。如果被隔离的主机是 SPM，这个 SPM 角色会被系统收回并分配给另外一台正常工作的主机。因为拥有 SPM 角色的主机是唯一一个可以修改数据域结构元数据的主机，所以如果作为 SPM 的主机没有配置隔离功能，当它出现故障时，环境中的所有需要修改数据域元数据的操作（如创建和销毁虚拟磁盘、进行快照、扩展逻辑卷等）都将无法进行。

当一个主机处于“无响应”状态时，在它上面运行的所有虚拟机也会处于“无响应”状态，而虚拟机对虚拟磁盘镜像操作所留下的“锁定”记录仍然会保留在主机上。这时，如果没有使用隔离功能，而直接在其它主机上重启那些无响应的虚拟机，并且虚拟机有写操作权限时，虚拟机原来的磁盘镜像中的数据可能会被破坏。

使用隔离功能可以避免这个问题的出现。当主机被重启后，以前的“锁定”记录会被释放。Red Hat Virtualization Manager 会使用一个隔离代理来确认出现问题的主机是否已经被重启。当 Manager 收到了主机已经重启成功的确认后，就可以在其它主机上运行原来在出现问题的主机上运行的虚拟机，而不会造成数据的破坏。隔离是实现高可用性虚拟机的基础，没有这个功能，高可用性虚拟机将无法在其它主机上运行。

当一个主机无响应时，Red Hat Virtualization Manager 会等待 30 秒的宽限期后决定是否进行其它操作，这可以避免因为主机的临时性错误造成的不必要的操作。当宽限期过后，主机仍然没有响应，Manager 就会自动启动隔离操作。Manager 使用电源管理设备的隔离代理来停止主机的运行；在确认主机已经停止后，再次启动主机，并确认主机已经被成功启动。当主机启动完成后，它会尝试重新加入到原来的集群中。如果主机的故障在启动后已被解决，它的状态会变为 **up**，并可以继续正常运行虚拟机。

## 4.5. Soft-Fencing 主机

有些时候，一个主机会因为无法预见的问题造成它处于无响应状态。此时尽管 VDSM 对所做出的请求无法响应，但依赖于 VDSM 的虚拟机仍然可以被访问。在这种情况下，重新启动 VDSM 就可能解决这个问题。

"SSH Soft Fencing" 是 Manager 试图通过 SSH 在一个没有响应的主机上重启 VDSM 的过程。如果 Manager 无法通过 SSH 重启 VDSM，而且配置了外部的隔离代理，则隔离操作将由外部的隔离代理进行处理。

要使用 soft-fencing over SSH 功能，主机必须配置并启用了隔离，一个有效的代理主机（数据中心中的另外一个主机，它的状态是 UP）必须存在。当 Manager 和主机的连接出现超时情况时，以下事件会发生：

1. 在网络出现第一次失败时，主机的状态变为 "connecting"。
2. Manager 然后会尝试 3 次向 VDSM 询问它的状态，或根据主机的负载等待一段时间。这个等待的时间是通过以下公式计算的： $\text{TimeoutToResetVdsInSeconds}$ （默认值是 60 秒）+  $[\text{DelayResetPerVmInSeconds}$ （默认值是 0.5 秒）] \* (在主机上运行的虚拟机的数量) +  $[\text{DelayResetForSpmlnSeconds}$ （默认值是 20 秒）] \* 1（如果主机是 SPM）或 0（如果主机不是 SPM）。为了留给 VDSM 最大的响应时间，Manager 会选择以上两个操作所需的最长时间。
3. 如果在所需要的间隔时间后主机还没有响应，**vdsmd restart** 命令会通过 SSH 执行。
4. 如果 **vdsmd restart** 命令无法在主机和 Manager 间重新创建连接，主机的状态将变为 **Non Responsive**，如果电源管理被配置，外部的隔离代理将会进行相应的隔离操作。



### 注意

Soft-fencing over SSH 可以在没有配置电源管理的主机上运行。这和一般的隔离（fencing）有所不同：一般的隔离只能在配置了电源管理的主机上运行。

## 4.6. 使用多个电源管理隔离代理

单一的隔离代理都会被看做为“主要的”代理。“次要的”隔离代理只有在有第二个代理存在时才有效，而多个代理可以是同一个类型，也可以是不同的类型。

如果一个主机上只有一个隔离代理，当这个代理出现问题时，主机在被手动重启前将会一直处于“无响应”的状态，而在它上面运行的虚拟机也将处于停止的状态。只有当主机被手工隔离后，这些虚拟机才会被迁移到另外的主机上运行。而如果一个主机上使用了多个隔离代理，当一个代理失败时，其它的代理就可以被使用，这样就可以提高隔离功能的稳定性。

当在主机上定义了两个代理时，这两个代理可以被配置为 *并行 (concurrent)* 或 *顺序 (sequential)* 使用：

- ✦ **并行 (Concurrent)**：要停止主机，主代理和从代理都需要对 Stop 命令进行响应；要启动主机，只需要一个代理对 Start 命令做出响应。
- ✦ **顺序 (Sequential)**：要停止或启动一个主机，主代理会被首先使用，如果使用主代理失败，从代理会被使用。

## 第 5 章 负载均衡、调度和迁移

### 5.1. 负载均衡、调度和迁移

一个单独主机所具有的硬件资源总是会有限制的，而这些硬件资源也会出现故障。要解决这些问题，我们可以把多个主机组成一个集群来在不同的主机间共享资源。Red Hat Virtualization 的环境使用负载均衡策略、调度和迁移来协调主机资源的使用情况。Red Hat Virtualization Manager 可以保证一个集群中的所有虚拟机不会只运行在一个主机上；另外，如果集群中出现某个主机的利用率非常低的情况，Manager 还会把这个主机上面的所有虚拟机迁移到其它主机上，从而可以关闭这个低利用率的主机来达到省电的目的。

当发生以下 3 个事件时，系统会检查环境中可用的资源：

- ✱ 启动虚拟机：系统会检查环境中的主机资源，并决定在哪个主机上运行这个虚拟机。
- ✱ 迁移虚拟机：系统会检查环境中的主机资源，并决定把虚拟机迁移到哪个主机上。
- ✱ 间隔一定时间：系统会定期检查环境中的主机资源，并决定每个主机是否符合预先设定的集群负载均衡策略。

当有效资源出现变化时，Manager 会根据集群的负载均衡策略来调度虚拟机迁移操作。下面会对负载均衡策略、调度和虚拟机迁移做详细的介绍。

### 5.2. 负载均衡策略

负载均衡策略是针对集群设置的。集群由一个或多个可以带有不同硬件参数和可用内存的主机组成，Red Hat Virtualization Manager 会根据集群的负载均衡策略来决定在集群中的哪个主机上运行虚拟机。另外，负载均衡策略还指定了 Manager 会在什么情况下把过高利用率主机上的虚拟机迁移到其它主机。

数据中心的每个集群都会每隔 1 分钟进行一次负载均衡处理。它会根据系统管理员预先为集群设定的负载均衡策略来决定哪些主机处于过度利用的状态；哪些主机的资源没有被充分利用；哪些主机可以运行从其它主机上迁移来的虚拟机。负载均衡策略选项包

括：`VM_Evenly_Distributed`、`Evenly_Distributed`、`Power_Saving` 和 `None`。

### 5.3. 负载均衡策略：`VM_Evenly_Distributed`

使用这个策略的集群会把需要运行的虚拟机根据数量平均运行在集群中的每个主机上。系统管理员需要设置每个主机上可以运行的“最多虚拟机数量”的值，如果某个主机上所运行的虚拟机数量超过了这个值，这个主机就被认为“过载（overload）”。另外，系统管理员还需要设备一个参数来指定最高利用率主机上所运行的虚拟机数量和最低利用率主机上所运行的虚拟机数量间的最大差值。以上的 2 个值决定了“虚拟机迁移阈值

（threshold）”的范围。当集群中的每个主机上所运行的虚拟机数据都在这个阈值范围内时，系统认为集群处于“负载均衡”的状态。除此之外，因为作为 SPM 的主机通常需要有较低的虚拟机负载，因此管理员还可以设置在 SPM 主机上被保留的虚拟机使用数量，这个值决定了 SPM 主机可以比其它主机少运行多少个虚拟机。当集群中的某个主机上所运行的虚拟机数量超过了“最多虚拟机数量”的值，并且集群中有最少一个主机所运行的虚拟机数量低于“虚拟机迁移阈值”的下限时，这个主机上的一个虚拟机就会被迁移到有最低利用率的主机上。如果迁移一个虚拟机后集群还没有处于“负载均衡”状态，系统会重复以上过程来迁移另外一个虚拟机，直到集群达到了“负载均衡”状态。

### 5.4. 负载均衡策略：`Evenly_Distributed`

这个策略会把新虚拟机运行在 CPU 或内存利用率最低的主机上。系统管理员可以设置一个“最大服务级别”值，这个值代表了集群中的主机所允许的最大 CPU 和内存利用率，如果超过了这个值，整个环境的性能就会下降。另外，系统管理员还可以设置一个时间值，它代表了在主机上的 CPU 和内存利用率超过了“最大服务级别”值多长时间后，Red Hat Virtualization Manager 才会对主机采取行动来解决这个问题。当这个时间值被超



过时，这个主机上的一个虚拟机会被迁移到集群中的 CPU 或内存利用率最低的一个主机上。如果迁移完成后，原来的主机的 CPU 和内存利用率仍然高于所限定的值，则重复以上步骤迁移它上面的另外一台主机，直到主机的 CPU 和内存利用率降到所限制的范围之内。

## 5.5. 负载均衡策略：Power\_Saving

这个策略会把新虚拟机运行在 CPU 和内存利用率最低的主机上。系统管理员可以设置一个“最大服务级别”值，这个值代表了集群中的主机所允许的最大 CPU 和内存利用率，如果主机的 CPU 和内存利用率超过了这个值，整个环境的性能就会下降。系统管理员还可以设置一个“最低服务级别”值，它代表了在主机的 CPU 和内存利用率低于这个值时，继续运行这个主机将会被认为从用电的角度来讲不符合“经济效益”。另外，系统管理员还需要为“最大服务级别”和“最低服务级别”设置一个时间值，它指定了主机的 CPU 和内存利用率低于“最低服务级别”值（或高于“最大服务级别”值）多长时间后，Red Hat Virtualization Manager 才会对主机采取行动来解决这个问题。当一个主机的 CPU 和内存利用率高于“最大服务级别”值，并且处于这个状态的时间超过了所设定的时间值时，这个主机上的一个虚拟机会被迁移到集群中的 CPU 和内存利用率最低的一个主机上。如果迁移完成后，原来主机的 CPU 和内存利用率仍然高于所限定的值，则重复以上步骤迁移它上面的另外一台主机，直到主机的 CPU 和内存利用率降到所限制的范围之内。当一个主机的 CPU 和内存利用率低于“最低服务级别”值，并且处于这个状态的时间超过了所设定的时间值时，这个主机上的所有虚拟机会在“最大服务级别”条件许可的情况下迁移到集群中的其它主机上。然后，Manager 会自动关闭这台主机。如果系统在以后的某个时间需要更多的主机资源时，这个被关闭的主机会被重新启动。

## 5.6. 负载均衡策略：None

如果没有选择任何负载均衡策略，新的虚拟机会在集群中的一个有可用内存，而且 CPU 利用率最低的主机上运行。其中的 CPU 利用率是通过一个算法对虚拟 CPU 的数量和 CPU 使用情况进行计算后得出的。如果选择使用这个选项，系统只有在要运行新虚拟机时才进行主机选择。另外，在主机负载增加时，系统也不会自动迁移虚拟机。

如果需要进行虚拟机迁移，系统管理员需要决定虚拟机要被迁移到的主机。另外，还可以使用 *固定* (*pinning*) 功能把虚拟机和某个主机相关联。使用固定功能可以防止虚拟机被自动迁移到其它主机。对于需要消耗大量硬件资源的环境，手工迁移是最好的选择。

## 5.7. 高可用性虚拟机资源保留

高可用性 (HA) 虚拟机资源保留策略可以使 Red Hat Virtualization Manager 监控集群资源的使用情况，从而可以保证在需要的时候为高可用性虚拟机提供有效的资源。Manager 可以把虚拟机标识为“高可用性”，在需要的时候，这些虚拟机可以在其它主机上重启。在高可用性虚拟机资源保留功能被启用时，Manager 会为高可用性虚拟机预留一些资源，当因为主机故障需要进行高可用虚拟机迁移时，可以保证集群中有足够的资源来完成迁移操作。

## 5.8. 调度

在 Red Hat Virtualization 中，调度 (scheduling) 是指 Red Hat Virtualization Manager 在集群中选择一个主机作为需要被迁移的虚拟机要迁移到的目标主机的过程。

作为一个可以运行新虚拟机或从其它主机上迁移来的虚拟机的主机，它需要有足够的空闲内存和 CPU 资源运行这些虚拟机。如果有多个主机都满足这个要求，系统会根据集群负载均衡策略来选择一个主机。例如，如果使用“Evenly\_Distributed”策略，Manager 会选择有最低 CPU 利用率的主机。如果使用“Power\_Saving”策略，在“最大服务级别”和“最低服务级别”范围内的、CPU 利用率最低的主机会被选择。另外，SPM 的状态也会影响主机的选择过程。一个没有 SPM 角色的主机比 SPM 主机有更高的被选择的可能性。例如，如果集群中有 SPM 主机和非 SPM 主机，第一个需要在集群中运行的虚拟机会在非 SPM 主机上运行。

## 5.9. 迁移

Red Hat Virtualization Manager 使用迁移来实现集群的负载均衡策略，当集群中的主机负载处于一定状态时，虚拟机迁移操作就会被执行，从而可以保证主机负载满足集群的负载均衡策略。另外，虚拟机迁移操作还可以被设置为当主机被隔离（fence）或被设置为维护模式时自动进行。当需要进行迁移时，Red Hat Virtualization Manager 会首先迁移 CPU 利用率最低的虚拟机（CPU 利用率是一个百分比值，在计算这个值时不会考虑内存和 I/O 的使用情况）。

在默认情况下，虚拟机迁移有以下限制：

- ✧ 每个虚拟机迁移的网络带宽被限制在 52 MiBps（megabyte 每秒）
- ✧ 迁移有超时限制，超时的计算方法是虚拟机内存的数量（以 GB 为单位）乘以 64 秒。
- ✧ 如果迁移进程没有任何进展的时间达到 240 秒，迁移会被终止。
- ✧ 当前，允许同时进行虚拟机迁移的数量被限制为每个主机上的 CPU 内核数量，但最多不能超过 2 个。

更多相关信息，请参阅 <https://access.redhat.com/solutions/744423>。

## 第 6 章 目录服务

### 6.1. 目录服务

用户在访问 Manager 的所有接口（用户门户、管理门户和 REST API）时，Red Hat Virtualization 平台需要使用目录服务来对用户进行身份验证和授权。Red Hat Virtualization 环境中的虚拟机也可以使用相同的目录服务来进行用户身份验证和授权，但需要对虚拟机进行相应的配置。当前 Red Hat Enterprise Manager 支持的目录服务包括 *Identity Management* (IdM)、*Red Hat Directory Server 9* (RHDS)、*Active Directory* (AD) 和 *OpenLDAP*。在进行以下操作时，Red Hat Enterprise Manager 需要使用目录服务：

- ✱ 用户登录（登录到用户门户、登录到管理门户、使用 REST API）。
- ✱ 查询并显示用户信息。
- ✱ 把 Manager 添加到域中。

身份验证（authentication）就是验证和识别一个对象，并保证这个对象所产生的数据完整性（data integrity）的过程。在身份验证中，主体（principal）被定义为需要被验证身份的对象；验证程序（verifier）被定义为请求主体身份验证的对象。对于 Red Hat Virtualization，Manager 是验证程序，而用户是主体。数据完整性保证了所接收到的数据和主体所产生的数据相同。

保密性（confidentiality）和授权（authorization）是和用户身份验证相关的两个概念。保密性是指受保护的数据不会被不应该访问它们的用户所访问，好的用户身份验证机制可以实现数据的保密性。授权决定了一个主体（principal）是否可以进行某个操作。Red Hat Virtualization 使用目录服务把用户和相关的角色进行关联，从而使用户获得相关的授权。授权的过程通常发生在用户身份验证之后，并可能需要本地或远程验证程序（verifier）的信息。

在安装的过程中，一个本地的内部域会被创建来管理 Red Hat Virtualization 环境。在安装完成后，用户可以根据需要添加其它的域。

### 6.2. 本地用户身份验证：内部域

Red Hat Virtualization Manager 在安装的过程中会创建一个功能有限的、内部的管理域。这个域和 AD 或 IdM 域不同，它是基于 Red Hat Virtualization PostgreSQL 数据库中的数据的，而不使用目录服务器所提供的目录服务。这个内部域只有一个用户：**admin@internal**。使用这个方法可以在没有外部目录服务器的情况下试用 Red Hat Virtualization；还可以使用这个管理员帐号对外部的目录服务进行故障排除。

使用 admin@internal 用户可以对一个虚拟环境进行初始的配置，如安装和批准主机、添加外部 AD 或 IdM 身份验证域、对外部域所提供的用户委托授权。

### 6.3. 使用 GSSAPI 进行远程身份验证

对于 Red Hat Virtualization，远程身份验证是指 Red Hat Virtualization Manager 在远程对用户进行身份验证。远程身份验证被用来验证 AD、IdM 或 RHDS 域中的用户（或使用这些用户的 API）到 Manager 的连接。系统管理员需要使用 **engine-manage-domains** 工具程序把 Red Hat Enterprise Virtualization Manager 配置为 RHDS、AD 或 IdM 域的一部分。Manager 需要一个 RHDS、AD 或 IdM 目录服务器中的帐号，这个帐号具有把系统加入到域中的权限。在域被添加后，Red Hat Virtualization Manager 就可以使用密码对域用户进行身份验证。Manager 使用 *Simple Authentication and Security Layer* (SASL) 平台中的 *Generic Security Services Application Program Interface* (GSSAPI) 来安全地验证用户的身份，并为用户赋予适当的权限。



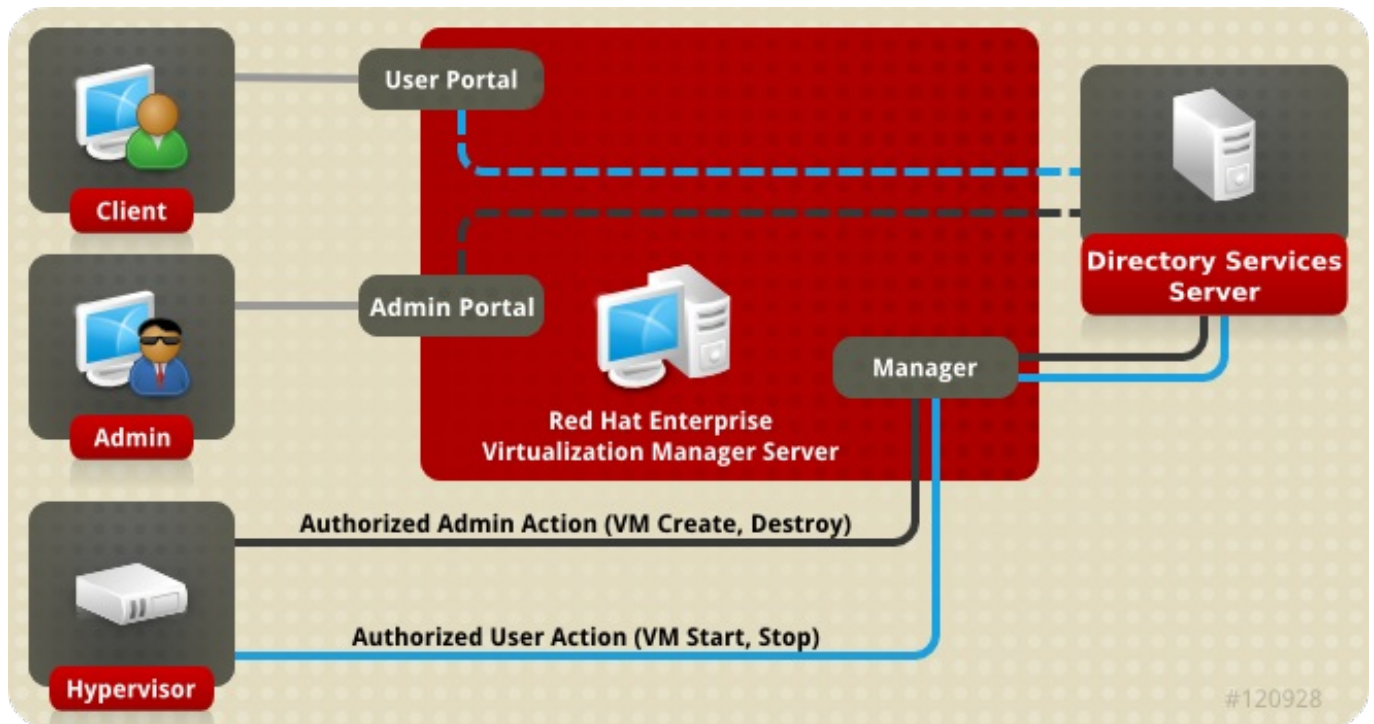


图 6.1. GSSAPI 身份验证

## 第 7 章 模板和虚拟机池

### 7.1. 模板和虚拟机池

Red Hat Virtualization 环境为管理员提供了两个用来简化为用户提供虚拟机操作的工具：*模板*（*template*）和*虚拟机池*（*pool*）。管理员可以通过使用模板来快速创建新的虚拟机。这个新虚拟机会基于一个已经存在的、并已经被配置好的虚拟机来创建，从而省去了手工安装操作系统和配置系统的步骤。模板功能对于需要创建多个相似虚拟机的环境非常有用。例如，您需要使用多个虚拟机作为 web 服务器。您可以首先在一台虚拟机上安装操作系统、安装 web 服务器软件并配置系统。然后，基于这个已经配置好的虚拟机创建一个模板。当您需要创建更多虚拟机来作为 web 服务器时，就可以使用这个模板来创建它们。

虚拟机池就是一组基于特定模板创建的虚拟机，这些虚拟机可以快速地提供给用户使用。对虚拟机池中的虚拟机的使用权限是在虚拟机池一级来设置的。如果一个用户有对某个虚拟机池的使用权限时，这个用户就有权使用这个虚拟机池中的任何虚拟机。因为用户每次从虚拟机池中获得的虚拟机可能并不是同一台虚拟机，所以虚拟机池并不适用于用户需要在虚拟机上保存数据的情况。虚拟机池适用于用户把数据保存在中央存储设备中的情况；或不需要存储新数据的情况。当虚拟机池创建完成后，组成这个虚拟机池的虚拟机会被创建，并处于“关闭”状态。当用户需要虚拟机时，虚拟机就会被启动。

### 7.2. 模板

要创建一个模板，管理员需要先创建一个虚拟机，在新虚拟机上安装所需的软件包并对新虚拟机进行配置。对于要被用来创建模板的虚拟机来讲，它的配置原则就是在被实施后就不需要对其进行大的改变。另外，管理员可以执行一个可选的（但推荐使用）的步骤：*泛化*（*generalization*）。泛化是指删除那些只与特定系统相关的、在不同的系统上会使用不同值的信息，如系统的用户名、密钥、时区。泛化对定制的配置不会有影响。Red Hat Enterprise Linux 虚拟机使用 **sys-unconfig** 进行泛化；Windows 虚拟机使用 **sys-prep** 进行泛化。如需了解更多关于在 Red Hat Enterprise Virtualization 环境中对 Windows 和 Linux 虚拟机进行泛化的信息，请参阅 *虚拟机管理指南* 中的 [Templates](#)。

当一个准备被用来创建模板的虚拟机被配置完成（如果需要，进行了泛化操作），并被停止运行后，管理员就可以基于这个虚拟机创建一个模板。在模板创建的过程中，模板所基于的虚拟机的磁盘镜像会被复制生成一个只读的镜像。这个只读的镜像就会作为所有今后基于这个模板所创建的虚拟机的基本磁盘镜像。换个角度来说，模板就是一个带有相关虚拟机硬件配置的、自定义的磁盘镜像。通过模板所创建的虚拟机的硬件配置可以被改变，例如，基于带有 1GB 内存的模板所创建的虚拟机可以被配置为带有 2GB 内存。但是，模板磁盘镜像本身不能被修改，这个因为对模板所做的修改将被应用到所有基于它所创建的虚拟机中。

当一个模板被创建后，您就可以使用它来作为创建多个虚拟机的基础。使用模板创建虚拟机有两种形式：*精简*（*thin*）模式和 *克隆*（*Clone*）模式。使用克隆模式所创建的虚拟机会具有所基于的模板基本镜像的完整的、可写的磁盘镜像备份。它的优点是所创建的虚拟机不再需要“依赖”所基于的模板（在所基于的模板不存在的情况下，仍然可以正常运行），而它的缺点是会使用更多的存储空间。使用精简模式创建的虚拟机会使用模板中的只读磁盘镜像作为基础磁盘镜像，并需要所基于的模板和基于这个模板所创建的虚拟机都位于同一个存储域中。每个虚拟机都会有一个可写的磁盘空间来保存添加和修改的数据。因为所有基于这个模板所创建的虚拟机都共享模板的只读基础磁盘镜像，所以使用这个模式创建虚拟机会节省存储空间。另外，因为共享的基础磁盘数据会被多次调用，所以它们会被保存在缓存中，这样就可以提高系统的性能。

### 7.3. 虚拟机池

虚拟机池可以快速地为用户提供相同的虚拟机（一般是虚拟桌面系统）。当一个有权利使用虚拟机池中的虚拟机的用户请求使用虚拟机时，用户的请求会被放置在一个“请求队列”中，系统会根据用户请求在“请求队列”中的位置来为用户提供一个可用的虚拟机。虚拟机池中的虚拟机不具有数据持久性，这意味着每次用户使用虚拟机池中的虚拟机时，这个虚拟机都处于它的基本状态，而用户上次使用虚拟机时对虚拟机所做的更改不会被保留。虚拟机池适用于用户的数据被存储在一个中央存储的情况。

虚拟机池是通过模板创建的。池中的每个虚拟机都共享一个后台的只读磁盘镜像，并使用一个临时的可写镜像保存在使用中需要保持的数据。位于虚拟机池中的虚拟机和其它虚拟机不同，用户在使用它们时所产生的数据

变化会在关闭虚拟机时被删除。这意味着虚拟机池所使用的存储较小（它只需要和模板相同的空间，再加上一些用来临时存储用户使用数据的存储空间）。使用存储池来提供虚拟机比为用户提供单独虚拟机要节省大量存储空间。

### 例 7.1. 虚拟机池使用实例

一家公司有 10 个技术支持员工，但在同一时间最多只会有 5 个技术支持人员进行工作。在这种情况下，可以使用一个虚拟机池（只需要包括 5 个虚拟机）来为技术支持人员提供虚拟机，而不需要为每个人都创建一个虚拟机（共需要创建 10 个虚拟机）。当一个技术支持人员开始工作时，可以从虚拟机池中获得一个虚拟机，在他完成工作时，把所使用的虚拟机返回到虚拟机池中。

## 第 8 章 虚拟机快照

### 8.1. 快照

快照就是一个存储功能，它允许管理员为一个虚拟机的操作系统、应用程序和数据在特定时间创建一个恢复点。快照会把虚拟机当前的磁盘镜像保存为一个 COW 卷，并可以在以后把虚拟机恢复到虚拟机创建快照时的状态。当快照创建完成后，一个新的 COW 层会在当前层上被创建，快照创建后的所有写操作都会发生在新的 COW 层上。

虚拟机的磁盘镜像实际上是由一个或多个卷组成的，而从虚拟机的角度来看，这些卷以一个单一的磁盘镜像形式所代表。

“COW 卷”和“COW 层”是同一个概念，它们可以被相互替代使用，但“层”更贴近于快照的本质。每创建一个快照，都将可以使管理员抛弃那些在创建快照后所做的改变，这和许多程序中所提供的 **Undo** 功能相似。



#### 注意

对标记为 **shareable** 的硬盘进行快照不被支持。另外，对基于 **Direct LUN** 连接的磁盘进行快照也不被支持。

快照包括 3 个主要操作：

- ✱ 创建：为一个虚拟机创建一个快照。
- ✱ 预览：预览一个快照，决定是否把系统恢复到创建这个快照时的状态。
- ✱ 删除：删除一个不再需要的恢复点（快照）。

如需了解更多与快照操作相关的信息，请参阅 *Red Hat Virtualization 虚拟机管理指南* 中的 [Snapshots](#)。

### 8.2. 实时快照

对标记为 **shareable** 的硬盘进行快照不被支持。另外，对基于 **Direct LUN** 连接的磁盘进行快照也不被支持。

所有其它没有正在进行克隆或迁移操作的虚拟机都可以在运行、暂停和停止的状态下进行快照。

当对一个虚拟机进行实时快照时，Manager 会要求 SPM 主机创建一个新的卷来为虚拟机使用。当新卷创建好后，Manager 调用 VDSM 来和运行虚拟机的主机上的 libvirt 和 qemu 进行交流，要求虚拟机的写操作在新卷上进行。如果虚拟机可以在新卷上进行写操作，则认为快照已经成功完成，虚拟机将不再在旧的卷中写数据。如果虚拟机无法在新卷上进行写操作，则认为快照操作失败，新的卷会被删除。

在虚拟机快照开始后，直到快照完成前，虚拟机需要对当前卷和新创建的卷都进行访问，因此这两个卷都需要允许进行读和写访问。

安装了带有静止（quiescing）功能的 guest 代理的虚拟机可以保证文件系统在不同快照间的一致性。已经注册的 Red Hat Enterprise Linux 虚拟机可以安装 **qemu-guest-agent** 来在进行快照前启用静止功能。

如果在进行快照时，虚拟机有支持静止功能的 guest 代理，VDSM 会使用 libvirt 和代理进行交流来准备快照。在实际进行快照前，没有完成的写操作会被完成，然后文件系统会被“冻结”。当快照操作完成后，libvirt 会把虚拟机的写操作切换到新的卷上，文件系统被“解冻”，对磁盘的写操作会被恢复。

所有的实时快照都会尝试使用静止功能。如果因为没有支持静止功能的 guest 代理而造成快照失败，实时快照会重新初始运行，但不会使用 `use-quiescing` 标志。当一个使用了静止功能文件系统的虚拟机使用快照恢复到以前状态时，虚拟机在启动时不会对文件系统进行检查。如果被恢复的虚拟机没有使用静止功能的文件系统，在使用快照恢复系统后，启动虚拟机时就需要对文件系统进行检查。

### 8.3. 创建快照

在 Red Hat Virtualization 中，第一次为一个虚拟机创建快照和以后为这个虚拟机创建后续快照的过程不同。虚拟机的第一个快照会保留镜像格式（QCOW2 或 RAW），它把当前存在的卷作为一个基础镜像。后续的快照只是一个附加的 COW 层，它只记录当前系统和前一个快照中的变化。

在 Red Hat Virtualization 中，一个虚拟机通常使用 RAW 磁盘镜像（除非在创建时使用了“精简（thin）”镜像，或用户指定使用 QCOW2 格式）。如 [图 8.1 “初始创建快照”](#) 所示，创建的快照将包括虚拟机的磁盘镜像作为后续快照的基本镜像。

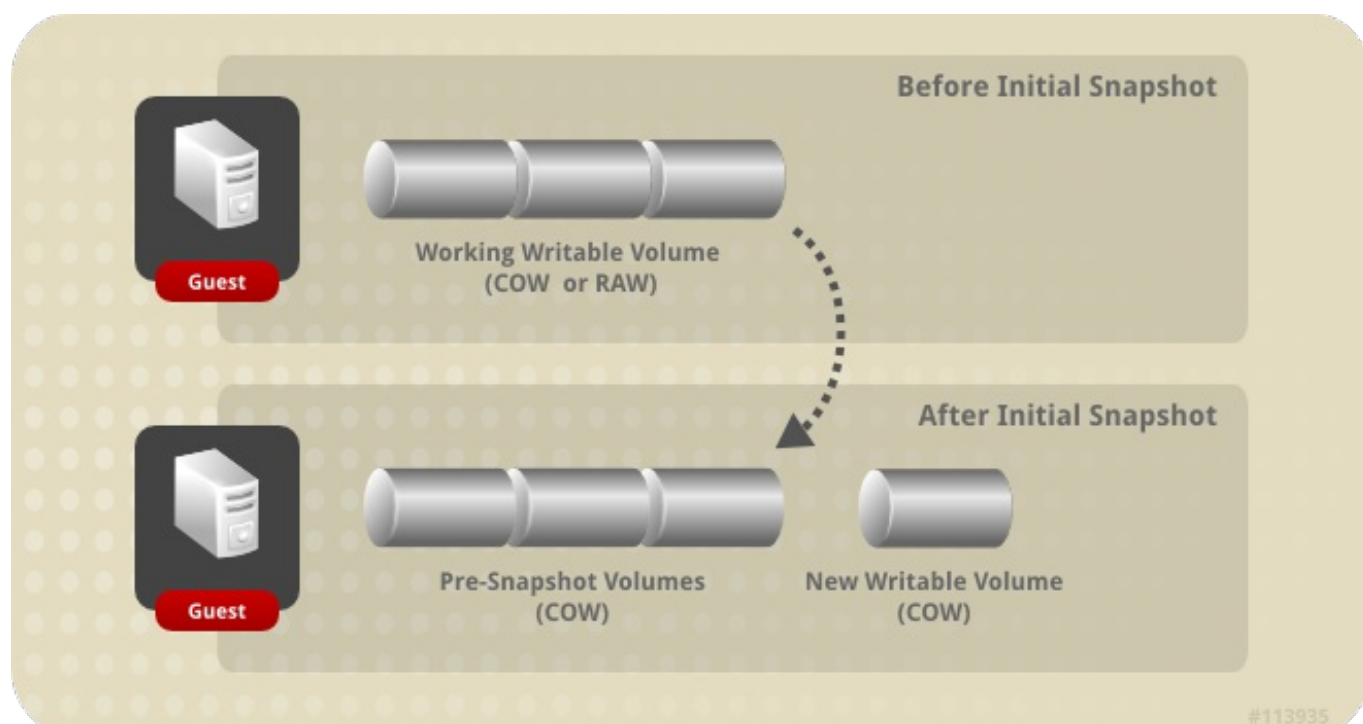


图 8.1. 初始创建快照

在第一个快照以后创建的后续快照将只创建一个新的 COW 卷，这个卷包括了当前系统和前一个快照间的变化。每个新的 COW 层在开始时都只包括 COW 元数据，而因为使用和操作虚拟机所产生的数据变化会被添加到这个新的 COW 层中。如果虚拟机需要修改前一个 COW 层的数据时，相应的数据会从前一层中读出，并把这些数据写到新的 COW 层中。虚拟机在定位数据时会以从最新到最老的顺序在各个 COW 中查找。



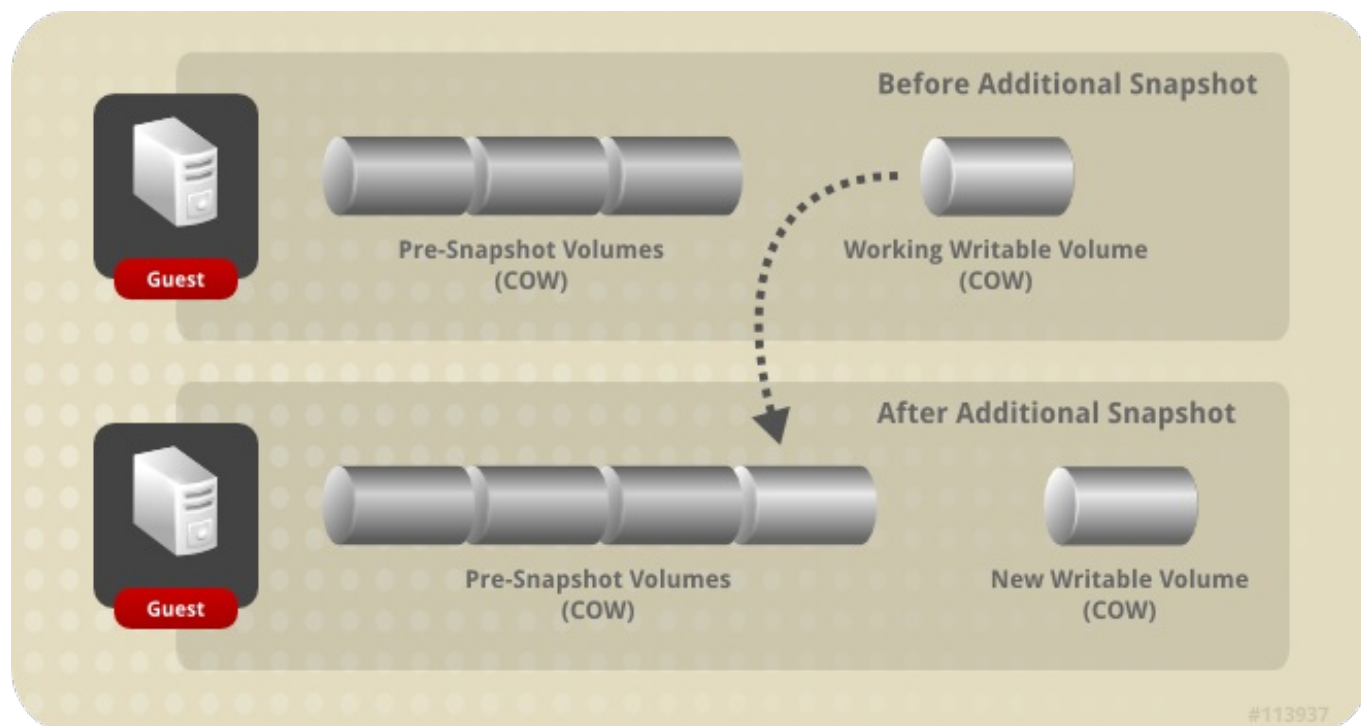


图 8.2. 创建后续快照

## 8.4. 预览快照

系统管理员可以预览以前创建的所有快照来决定把虚拟机恢复到什么状态。

管理员可以在一台虚拟机的所有快照列表选择一个快照来查看它的内容。如 [图 8.3 “预览快照”](#) 所示，每个快照都被保存为一个 COW 卷，当它被预览时，这个快照会被复制到一个新的预览层上。虚拟机将会和预览层进行交流，而不是直接访问实际的快照。

在管理员预览快照后，可以使用这个快照来把虚拟机恢复到快照的状态。如果管理员使用快照进行恢复系统后，虚拟机将会被关联到预览层。

在快照预览完成后，管理员可以选择 **Undo** 来删除在预览过程中创建的预览层。这时虽然预览层会被删除，原来保存快照的层还会保留。

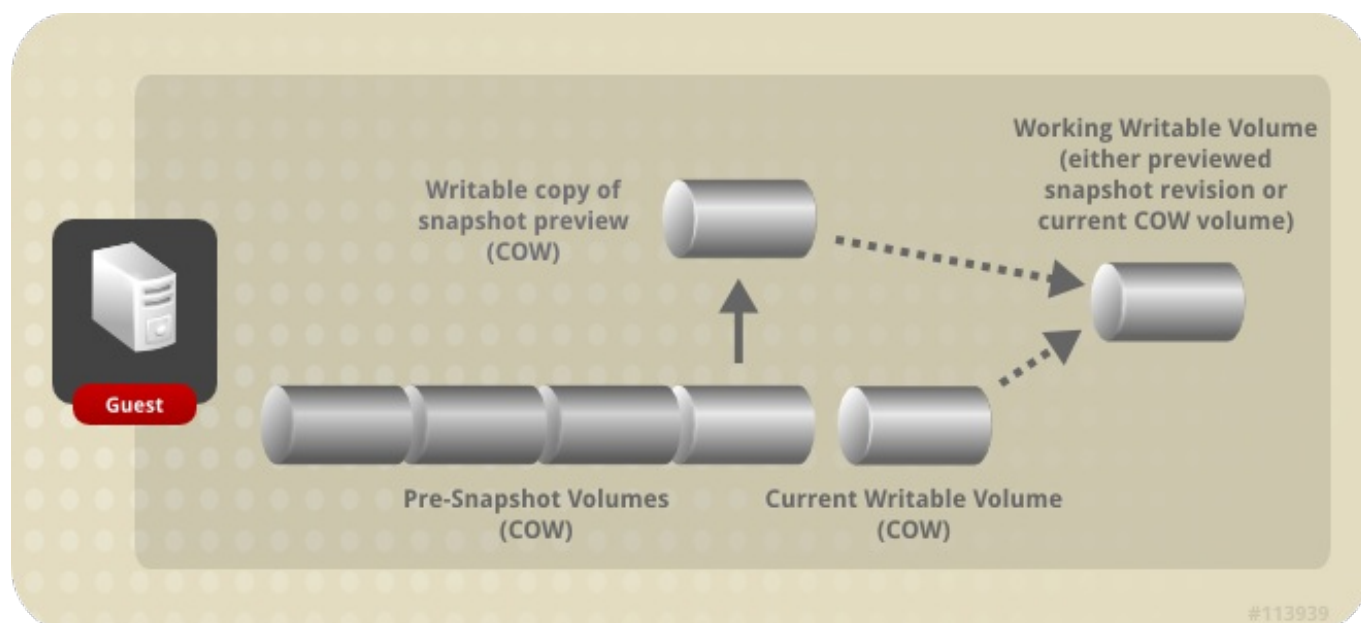


图 8.3. 预览快照



## 8.5. 删除快照

当一个或多个快照不再需要时，您可以删除它们。删除快照后，您将无法把虚拟机恢复到这些快照所包括的时间点上。删除快照并不一定会获得更多的可用存储空间，而这些快照的数据也不一定会被实际删除。例如，您的虚拟机有 5 个快照，如果您删除了第 3 个快照，第 3 个快照中的数据可能仍然会存在在系统中，因为第 4 和第 5 个快照可能会需要这些数据。一般情况下，删除快照通常可以提高虚拟机的性能。

当选择了一个要被删除的快照时，QEMU 会创建一个和要被删除的快照大小相同的新逻辑卷。被删除的快照会和它随后的那个快照进行合并，而新建逻辑卷的大小会被扩充来保存快照合并的结果。如果被删除的快照和它随后的那个快照没有重叠的内容，这个新建逻辑卷的大小将和被合并的两个快照的大小总和相同。当合并完成后，随后的那个快照会被改名，并被标记为已被删除，而用来保存合并结果的逻辑卷会使用它的名字来替代这个快照。最终删除的结果是选择被删除的快照和它随后的那个快照都会标记为已被删除，而这两个快照的合并结果会作为一个快照来取代它们的位置。

例如，快照 **Delete\_snapshot** 的大小是 200GB，它随后的那个快照 (**Next\_snapshot**) 的大小是 100GB。**Delete\_snapshot** 被删除，一个新的逻辑卷（临时命名为 **Snapshot\_merge**）会被创建，它的初始大小是 200GB。然后，**Snapshot\_merge** 的大小会被扩展到 300GB 来保存 **Delete\_snapshot** 和 **Next\_snapshot** 的合并结果。随后，**Next\_snapshot** 会被改名为 **Delete\_me\_too\_snapshot**，而 **Snapshot\_merge** 被重新命名为 **Next\_snapshot**。最后，**Delete\_snapshot** 和 **Delete\_me\_too\_snapshot** 都被删除。

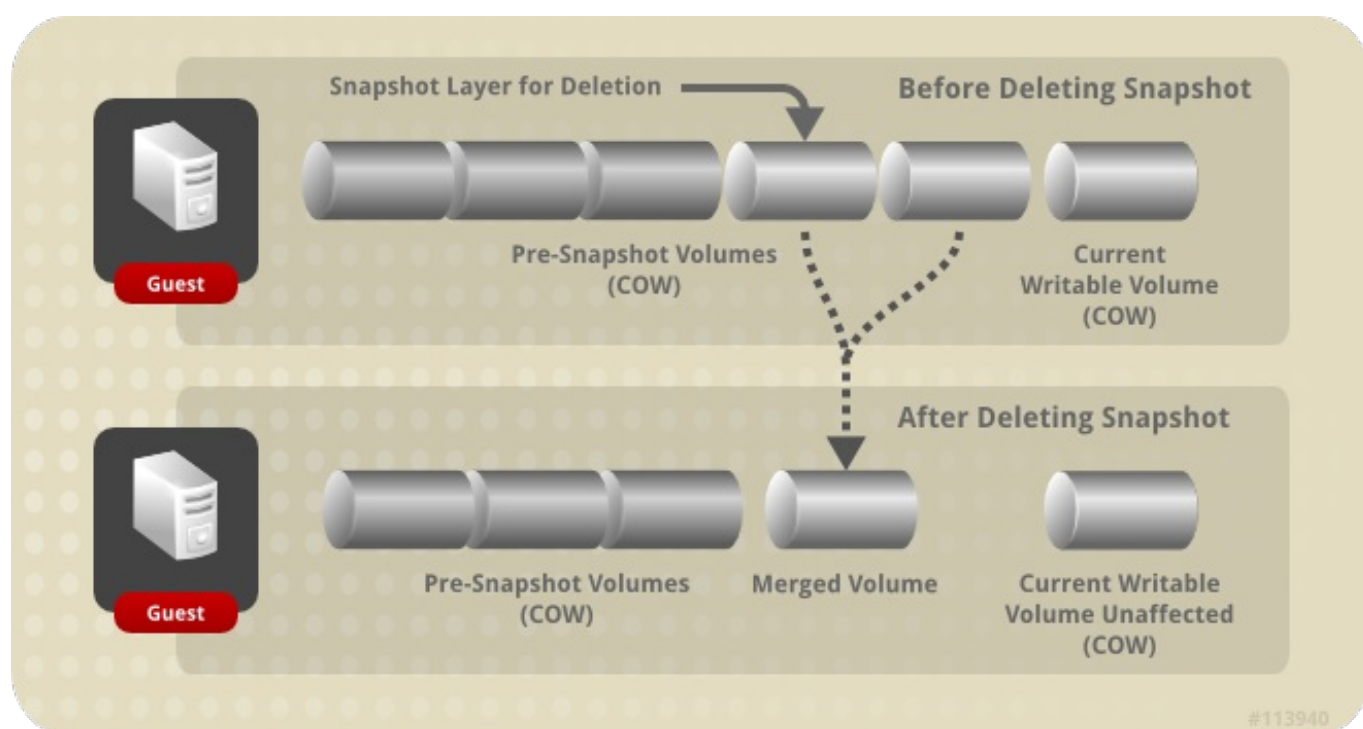


图 8.4. 删除快照

从正在运行的虚拟机上删除快照的逻辑和从关闭的虚拟机上删除快照的逻辑有一点不同。实时快照删除会被作为一个异步的块任务处理，VDSM 会为虚拟机在恢复文件中维护一个操作记录，从而使此任务可以被跟踪，即使在进行操作期间 VDSM 被重启或虚拟机被关闭时任务也可以被跟踪。当操作开始后，正被删除的快照将不能被预览，或作为一个恢复点（即使删除操作失败或被中断）。活跃(layer)被合并到它的上一层的操作被分为两个阶段 -- 数据被从活跃层复制到它的上一层，磁盘写操作被镜像到活跃层和它的上一层。最后，当镜像中的数据被合并到它的上一层快照中，VDSM 把镜像链进行同步后，删除操作就被认为已经完成了。

## 第 9 章 硬件驱动和设备

### 9.1. 虚拟硬件

Red Hat Virtualization 为虚拟机提供了 3 种不同类型的系统设备，它们都以物理硬件设备的形式出现在虚拟机上，而这些设备的驱动会以不同的形式工作。

#### 模拟设备

模拟设备（有时称为*虚拟设备*）是完全由软件实现的设备。*模拟设备驱动*被作为一个在主机（用来管理后台源设备）操作系统和虚拟机操作系统间的“翻译”层，对虚拟设备的指令会通过虚拟机管理器（hypervisor）来进行“翻译”。任何和模拟设备类型相同、并可以被 Linux 内核识别的设备都可以作为虚拟驱动的后台源设备。

#### 准虚拟设备（Para-virtualized Device）

准虚拟设备需要在虚拟机上安装设备驱动来建立一个和主机上的虚拟机管理器进行交流的接口，它可以使那些通常需要占用大量资源的操作（如磁盘 I/O 操作）在虚拟环境外进行。使用这种方式，可以减少对虚拟机环境资源的占用，从而使在虚拟机上运行的操作系统的性能更接近于直接运行在物理机上的操作系统的性能。

#### 物理共享的设备

特定的硬件平台允许虚拟机直接访问一些硬件设备和组件，这在虚拟环境中被称为*透传*（*passthrough*）或*设备分配*（*device assignment*）。透传可以使设备象物理接连接到虚拟机上的设备一样使用。

### 9.2. 在 Red Hat Virtualization 环境中的固定设备地址

虚拟硬件的 PCI 地址记录在 ovirt-engine 数据库中，因此它们的 PCI 地址是固定的。

在虚拟机被创建时，**QEMU** 会为虚拟硬件设备配置 PCI 地址，并通过 **libvirt** 报告给 **VDSM**。**VDSM** 把这个信息传递给 Manager，Manager 把地址信息保存在 ovirt-engine 数据库中。

当一个虚拟机启动时，Manager 会从数据库中读相应的设备地址信息，并把它传递给 **VDSM**。然后，**VDSM** 把这些设备地址信息再传递给 **libvirt**，从而使虚拟机可以使用这些从数据库中获得的 PCI 设备地址。

当一个设备被从虚拟机上删除时，包括 PCI 地址在内的、与这个虚拟机相关的信息都会被删除。如果需要添加一个新的设备来替代被删除的设备时，**QEMU** 会为新设备提供一个新的 PCI 地址。

### 9.3. 中央处理器（CPU）

一个集群中的每个主机都会有一定数量的*虚拟 CPU*（*vCPU*）。这些虚拟 CPU 可以被在这个主机上运行的虚拟机使用。Red Hat Virtualization Manager 在创建主机所在的集群时需要指定虚拟 CPU 的类型。一个集群中的虚拟 CPU 类型必须是相同的。

每个有效虚拟 CPU 类型所具有的功能特征是由它们所基于的、同名的物理 CPU 所决定的。对于虚拟机操作系统而言，虚拟 CPU 和物理 CPU 无法被区分。



## 注意

对 x2APIC 的支持：

由 Red Hat Enterprise Linux 7 主机所提供的所有虚拟 CPU 类型都支持 x2APIC。它提供了一个高级可编程中断控制器 - *Advanced Programmable Interrupt Controller (APIC)* 来更好地处理硬件中断。

## 9.4. 系统设备

系统设备对虚拟机是必须的，它们不能从虚拟机上删除。每个附加到虚拟机上的系统设备都会占用一个 PCI 插槽。默认的系统设备是：

- » 主桥 (host bridge)
- » ISA 桥和 USB 桥 (USB 桥和 ISA 桥是相同的设备)
- » 图形卡 (使用 Cirrus 或 qxl 驱动)
- » 内存气球设备 (memory balloon device)

## 9.5. 网络设备

Red Hat Virtualization 可以为虚拟机提供 3 种不同类型的网络接口控制器。当创建虚拟机时，网络接口控制器的类型会被指定，并且可以使用 Red Hat Virtualization Manager 修改它。

- » **e1000** 网络接口控制器为虚拟机提供了一个虚拟的 Intel PRO/1000 (e1000) 设备。
- » **virtio** 网络接口控制器为虚拟机提供了一个准虚拟化的网络设备。
- » **rtl8139** 网络接口控制器为虚拟机提供了一个虚拟的 **Realtek Semiconductor Corp RTL8139** 设备。

一个虚拟机上可以有多个网络接口控制器，而每个控制器都会占用虚拟机上的一个 PCI 插槽。

## 9.6. 图形设备

系统提供了两个模拟图形设备。这些设备可以通过使用 SPICE 协议或使用 VNC 进行连接。

- » **ac97** 模拟一个 **Cirrus CLGD 5446 PCI VGA** 卡。
- » **vga** 模拟一个带有 **Bochs VESA** 扩展的 **VGA** 卡 (硬件层，包括所有非标准模式)。

## 9.7. 存储设备

存储设备和存储池可以使用块设备驱动把存储设备附加到虚拟机上。这里请注意，存储驱动并不是存储设备，它被用来把后台的存储设备、文件或存储池卷附加到虚拟机上。而后台的存储设备可以是任何被支持的存储设备、文件和存储池卷。

- » **IDE** 驱动为虚拟机提供了一个模拟的块设备。模拟的 **IDE** 驱动可以为每台虚拟机提供最多 4 个虚拟 **IDE** 磁盘和虚拟 **IDE CD-ROM** 设备。模拟的 **IDE** 驱动也为虚拟机提供虚拟 **DVD-ROM** 设备。

- ✧ **VirtIO** 驱动为虚拟机提供了一个准虚拟化的块设备。准虚拟化块设备驱动就是所有被 hypervisor 支持的、被添加到虚拟机上的设备（不包括软盘设备，软盘设备需要被模拟）驱动。

## 9.8. 音响设备

模拟的音响设备包括：

- ✧ **ac97** 模拟一个 **Intel 82801AA AC97 Audio** 兼容声卡。
- ✧ **es1370** 模拟一个 **ENSONIQ AudioPCI ES1370** 声卡。

## 9.9. 串行驱动

准虚拟化串行驱动（**virtio-serial**）是一个使用字节流的字符流驱动。它为主机的用户空间和虚拟机的用户空间提供了一个简单的交流接口。

## 9.10. “气球”（balloon）驱动

“气球”（balloon）驱动允许虚拟机通知 hypervisor 它们需要使用的内存数量。通过使用这个驱动，主机可以有效地为虚拟机分配内存，并可以把虚拟机上的可用内存分配给其它虚拟机和进程。

使用“气球”驱动的虚拟机可以把它的某些内存段标记为“not in use”，hypervisor 将可以把这些内存分配给运行在这个主机上的其它虚拟机和进程（气球充气）。当原来的虚拟机需要这些内存时，hypervisor 可以重新把内存分配给这个虚拟机（气球放气）。

## 第 10 章 最小的配置要求和技术限制

### 10.1. 最小的硬件配置要求和限制

Red Hat Virtualization 环境有一系列的物理限制和逻辑限制。如果您的系统配置超出了这些限制，它们将不被支持。

### 10.2. 数据中心的限制

在一个虚拟化环境中，数据中心是所有资源的最高一级容器（container）。包括在每个数据中心中的资源会有以下限制。

表 10.1. 数据中心的限制

资源项	限制
存储域的数量	✱ 我们推荐每个数据中心最少包括 2 个存储域：一个必需的数据域，以及一个推荐使用的 ISO 存储域。
主机的数量	✱ 每个数据中心最多支持 200 个主机。

### 10.3. 集群限制

群集由一组物理主机组成，它被当作虚拟机的资源池。群集中的主机共享相同的网络基础结构和存储空间。它们组成了一个迁移域，里面的虚拟机可以在主机间移动。为了保证稳定性，集群有它自己的限制。

- ✱ 所有可管理的 hypervisor 都必须包括在某个集群中。
- ✱ 包括在同一个集群中的 hypervisor 必须有相同的 CPU 类型。Intel 和 AMD CPU 不能在同一个集群中共存。



#### 注意

如需了解更多与集群相关的信息，请参阅 *管理指南* 中的 [Clusters](#)。

### 10.4. 存储域限制

存储域为虚拟机磁盘镜像和 ISO 镜像，以及导入和导出虚拟机操作提供存储空间。一个数据中心可以包括多个存储域，而每个存储域都会有它自己的限制和推荐的设置。

表 10.2. 存储域限制

项	限制
---	----

项	限制
存储类型	<p>支持的存储类型包括：</p> <ul style="list-style-type: none"> <li>光纤通道协议 (Fibre Channel Protocol - FCP)</li> <li>内部小型计算机接口 (Internet Small Computer System Interface - iSCSI)</li> <li>网络文件系统 (Network File System - NFS)</li> <li>POSIX 兼容的文件系统 (POSIX)</li> <li>Red Hat Gluster Storage (GlusterFS)</li> </ul> <p>Red Hat Virtualization 4.0 中新的 ISO 和导出域可以由任何基于文件的存储 (NFS、Posix 或 GlusterFS) 提供。</p>
Logical Unit Numbers (LUN)	由 iSCSI 或 FCP 所提供的存储域最多只能包括 300 个 LUN。
逻辑卷 (Logical Volume - LV)	<p>在 Red Hat Virtualization 中，逻辑卷代表了虚拟机、模板和虚拟机快照所使用的虚拟磁盘。</p> <p>对于 iSCSI 或 FCP，我们推荐在每个存储域中最多包括 350 个逻辑卷。如果一个存储域中包括了多于 350 个逻辑卷，我们推荐您把它们分到不同的存储域中。</p> <p>以上限制是由 LVM 元数据的大小造成的。当逻辑卷的数量增加时，与这些元数据相关的 LVM 元数据的大小也会增加。当元数据超过 1 MB 时，相关操作（如创建新磁盘、缩小快照大小、为使用“thinly provisioning”的逻辑卷进行 lvextend 操作）会需要长时间来完成。</p> <p>如需了解更多与逻辑卷相关的信息，请参阅 <a href="https://access.redhat.com/solutions/441203">https://access.redhat.com/solutions/441203</a>。</p>



## 注意

如需了解更多与存储域相关的信息，请参阅 *管理指南* 中的 [Storage](#)。

## 10.5. Red Hat Virtualization Manager 限制

Red Hat Virtualization Manager 服务器必须运行在 Red Hat Enterprise Linux 7 上，另外，它对硬件配置也有一定的要求。

表 10.3. Red Hat Virtualization Manager 限制

项	要求
内存	<ul style="list-style-type: none"> <li>需要最少 4GB 内存。</li> </ul>



项	要求
PCI 设备	<ul style="list-style-type: none"> <li>➤ 推荐最少使用一个最小带宽为 1 Gbps 的网络控制器。</li> </ul>
存储	<ul style="list-style-type: none"> <li>➤ 推荐最少具有 25GB 可用本地磁盘空间。</li> </ul>



### 注意

如需了解更多关于 Red Hat Virtualization Manager 的信息，请参阅[安装指南](#)。

## 10.6. Hypervisor 配置要求

Red Hat Virtualization Host (RHVH) 对硬件有一定要求和支持限制，而 Red Hat Enterprise Linux 主机所需的存储空间会根据情况有所不同，但它们会比 Red Hat Virtualization Host 的存储配置要求更高。

表 10.4. Red Hat Virtualization Hypervisor 硬件配置要求和限制。

项	要求和限制
CPU	<p>最少需要一个物理 CPU。Red Hat Virtualization 支持在虚拟主机中使用的 CPU 型号包括：</p> <ul style="list-style-type: none"> <li>➤ AMD Opteron G1</li> <li>➤ AMD Opteron G2</li> <li>➤ AMD Opteron G3</li> <li>➤ AMD Opteron G4</li> <li>➤ AMD Opteron G5</li> <li>➤ Intel Conroe</li> <li>➤ Intel Penryn</li> <li>➤ Intel Nehalem</li> <li>➤ Intel Westmere</li> <li>➤ Intel Haswell</li> <li>➤ Intel SandyBridge 系列</li> <li>➤ IBM POWER 8</li> </ul> <p>所有 CPU 都必须支持 Intel® 64 或者 AMD64 CPU 扩展，并启用 AMD-V™ 或者 Intel VT® 硬件虚拟化扩展。还要求支持 <b>No eXecute</b> 标签 (NX)。</p>

项	要求和限制
内存	<p>每个虚拟机所需内存的具体数量取决于以下因素：</p> <ul style="list-style-type: none"> <li>❖ 虚拟机操作系统对内存的要求</li> <li>❖ 虚拟机上运行的应用程序对内存的要求</li> <li>❖ 对虚拟机内存的使用情况。</li> </ul> <p>另外，KVM 可以为虚拟机“过度分配（over-commit）”物理内存。这是通过只为虚拟机提供它们正在需要使用的内存，而把其它没有被使用的内存移到交换区中来实现的。</p> <p>如需了解更多与虚拟机所支持的最大和最小内存数量相关的信息，请参阅 <a href="https://access.redhat.com/articles/rhel-limits">https://access.redhat.com/articles/rhel-limits</a>。</p>
存储	<p>主机所需的最少内部存储的数量是以下存储要求的总和：</p> <ul style="list-style-type: none"> <li>❖ root (/) 分区最少需要 6 GB 存储空间。</li> <li>❖ /boot 分区最少需要 1 GB 存储空间。</li> <li>❖ /var 分区最少需要 15 GB 存储空间。对于自承载引擎（self-hosted engine）部署，这个分区最少需要 60 GB。</li> <li>❖ 交换分区需要最少 8MB 存储，您在设定它的具体值时需要考虑这个主机的实际情况，以及在环境中可能出现的“内存过度分配”的情况。如需了解更多相关信息，请参阅 <a href="https://access.redhat.com/solutions/15244">https://access.redhat.com/solutions/15244</a>。</li> </ul> <p>请注意，以上是主机对存储空间的最底要求。我们推荐您使用默认的存储设置，这会需要更多的存储空间。</p>
PCI 设备	<p>推荐最少使用一个最小带宽为 1Gbps 的网络控制器。</p>



## 重要

在 Red Hat Virtualization Host 引导过程中可能会出现以下警告信息：

```
Virtualization hardware is unavailable.
(No virtualization hardware was detected on this system)
```

如果出现以上信息，则说明您的 CPU 不包括虚拟化扩展功能，或虚拟化扩展功能被禁用。请确定您的 CPU 支持虚拟化扩展，而且这个扩展在系统的 BIOS 中被启用。

使用以下方法检查 CPU 是否有虚拟化扩展功能，以及这个功能是否已经被启用：

- ✧ 在主机引导页面中按任意键，并选择列表中的 **Boot** 或 **Boot with serial console** 项。按 **Tab** 键编辑所选项的内核参数。确定在最后一个内核参数后有一个空格，并添加了 **rescue** 参数。
- ✧ 点 **Enter** 键把系统启动到 rescue 模式。
- ✧ 当系统提示符出现后，运行以下命令确定您的处理器是否有虚拟化扩展，以及是否启用了虚拟化扩展：

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

如果有任何结果输出，那么该处理器就可以进行硬件虚拟化。如果没有结果，您的处理器也仍有可能支持硬件虚拟化。在有些情况下生产商会在 BIOS 中禁用虚拟化扩展。请查看系统 BIOS 以及生产商提供的主板手册。

- ✧ 另外，请检查 **kvm** 模块是否被内核加载：

```
# lsmod | grep kvm
```

如果以上输出包括 **kvm\_intel** 或 **kvm\_amd**，**kvm** 硬件虚拟化模块被加载，这就意味着您的系统满足要求。

## 10.7. 虚拟机硬件配置要求和限制

运行在 Red Hat Virtualization Host (RHVH) 上的虚拟机有以下硬件配置要求和限制：

表 10.5. 虚拟硬件

项	要求和限制
CPU	在 Red Hat Enterprise Linux 7 中，每个虚拟机最多支持 240 个虚拟 CPU。
内存	<p>不同的虚拟机对内存有不同的要求。每台虚拟机具体所需的内存由虚拟机的操作系统以及虚拟机操作负载所决定。</p> <p>如需了解更多与虚拟机所支持的最大和最小内存数量相关的信息，请参阅 <a href="https://access.redhat.com/articles/rhel-kvm-limits">https://access.redhat.com/articles/rhel-kvm-limits</a>。</p>

项	要求和限制
PCI 设备	每个虚拟机最多可以支持 31 个虚拟 PCI 设备。一些系统设备（其它包括一些必需的系统设备）会占用所限制的设备数量，如 PCI 主机桥、ISA 桥、USB 桥、板桥（board bridge），图形卡以及 IDE 或 VirtIO 块设备。
存储	一个虚拟机可以最多支持 28 个虚拟存储设备（3 个 IDE 设备和 25 个 Virtio 设备）。

## 10.8. SPICE 的限制

SPICE 当前支持的最大分辨率是 2560x1600。

## 10.9. 额外参考信息

这些额外的文档没有包括在 Red Hat Virtualization 文档套件中，但它们可以为系统管理员管理 Red Hat Enterprise Virtualization 环境提供有用的信息。这些文档可以通过 <https://access.redhat.com/documentation/en-US> 获得。

### ***Red Hat Enterprise Linux - 系统管理员指南***

提供了部署、配置和管理 Red Hat Enterprise Linux 的信息。

### ***Red Hat Enterprise Linux - DM-Multipath Guide***

提供了在 Red Hat Enterprise Linux 中使用 Device-Mapper Multipathing 的信息。

### ***Red Hat Enterprise Linux - Installation Guide***

提供了关于安装 Red Hat Enterprise Linux 的信息。

### ***Red Hat Enterprise Linux - Storage Administration Guide***

提供了在 Red Hat Enterprise Linux 中管理存储设备和文件系统的信息。

### ***Red Hat Enterprise Linux - 虚拟化部署和管理指南***

提供了在 Red Hat Enterprise Linux 上安装、配置、管理和维护虚拟资源的信息。