第 3 章

实现嵌套虚拟化

嵌套虚拟化是指在一个虚拟机之中再运行另外一个虚拟机。由于嵌套虚拟化有一些功能上的限制，所以通常不建议在生产环境中使用，它主要用于开发和测试。有多种嵌套虚拟化的方法，本章将介绍通过 KVM 实现嵌套虚拟化。

本章要点

- ❑ 嵌套虚拟化的原理。
- ❑ L1 级别宿主机的准备。
- ❑ L2 级别 KVM 宿主机的配置。
- ❑ L2 级别 VMware ESXi 宿主机的配置。
- ❑ L2 级别 Microsoft Hyper-V 宿主机的配置。

3.1 嵌套虚拟化的原理

为了保证虚拟化平台的安全和稳定，默认情况下 L1 级别（也称为第一层）的 Hypervisor 会阻止其他软件使用 CPU 的虚拟化功能，所以其上的虚拟机就无法再充当 Hypervisor 了。

如果在 L1 级别的 Hypervisor 上启动了对嵌套虚拟化的支持，它就会向其虚拟机公开硬件虚拟化扩展。这些虚拟机可以充当 L2 级别的 Hypervisor，安装并运行自己的虚拟机，如图 3-1 所示。图中箭头表示硬件虚拟化扩展。

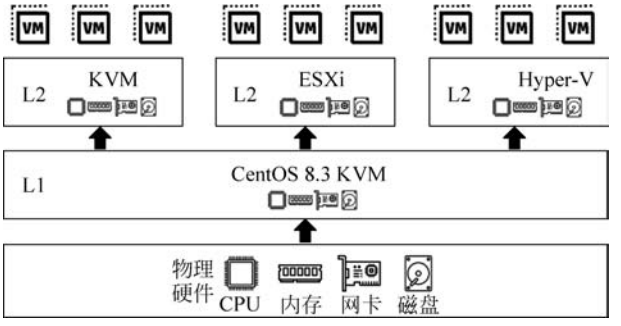

图 3-1　嵌套虚拟化原理

要实现嵌套虚拟化，需要 L1 和 L2 级别都支持 Hypervisor。RHEL/CentOS 8.2 及更高版本全面支持 Intel CPU 上的嵌套虚拟化功能。Red Hat 公司当前仅在 Intel CPU 嵌套虚拟化提供技术支持。在 AMD、IBM POWER9 和 IBM Z 系统上的嵌套虚拟化仅作为技术预览提供，因此 Red Hat 公司不提供支持服务。

早期的 Linux 发行版本例如 CentOS 7.2，无法实现 Microsoft Hyper-V 2012 和 2016 的嵌套虚拟化。随着技术的发展，RHEL/CentOS 8.3 可以很好地支持 Microsoft Hyper-V 2019 的嵌套虚拟化。

3.2 L1 级别宿主机的准备

嵌套虚拟化对运行 L1 级别 Hypervisor 宿主机的硬件要求比较高。下面的实验中使用一台 HP Z420 工作站，通过 lshw 命令查看其硬件信息，示例命令如下：

```
#lshw -short
H/W path   Class      Description
================================
           system     HP Z420 Workstation (LJ449AV)
/0         bus        1589
/0/0       memory     64KiB BIOS
/0/4       processor  Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz
/0/4/5     memory     512KiB L1 cache
/0/4/6     memory     2MiB L2 cache
/0/4/7     memory     20MiB L3 cache
/0/44      memory     32GiB System Memory
/0/44/0    memory     8GiB DIMM DDR3 Synchronous Registered (Buffered) 1333 MHz (0.8 ns)
/0/44/1    memory     DIMM Synchronous [empty]
/0/44/2    memory     8GiB DIMM DDR3 Synchronous Registered (Buffered) 1333 MHz (0.8 ns)
/0/44/3    memory     DIMM Synchronous [empty]
/0/44/4    memory     DIMM Synchronous [empty]
/0/44/5    memory     8GiB DIMM DDR3 Synchronous Registered (Buffered) 1333 MHz (0.8 ns)
/0/44/6    memory     DIMM Synchronous [empty]
/0/44/7    memory     8GiB DIMM DDR3 Synchronous Registered (Buffered) 1333 MHz (0.8 ns)
...
```

这台宿主机包括一个 16 核的 Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz，32GB 内存。在其上安装 CentOS 8.3 并进行升级，升级后的版本信息如下：

```
#cat /etc/redhat-release
CentOS Linux release 8.3.2011

#uname -a
Linux localhost.localdomain 4.18.0-240.el8.x86_64 #1 SMP Fri Sep 25 19:48:47 UTC 2020 x86_
64 x86_64 x86_64 GNU/Linux
```

HP Z420 工作站的 BIOS 默认启用了虚拟化支持，可以通过检查/proc/cpuinfo 是否包含 vmx 和 ept 标志来确认，示例命令如下：

```
#cat /proc/cpuinfo | egrep '(vmx|ept)'
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon
pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_
cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer
aes xsave avx lahf_lm epb pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid
xsaveopt dtherm ida arat pln pts md_clear flush_l1d
...
```

检查内核参数是否启用了嵌套虚拟化的支持，示例命令如下：

```
#cat /sys/module/kvm_intel/parameters/nested
0
```

如果命令返回 1 或 Y，则表示已经启用了该功能。如果命令返回 0 或 N，则表示未启用。

有两种方法启用对嵌套虚拟化的支持。

第 1 种方法是使用 modprobe 命令进行临时启用，示例命令如下：

```
#modprobe -r kvm_intel

#modprobe kvm_intel nested=1
```

先通过 modprobe 命令卸载 kvm_intel 模块，然后使用选项 nested=1 启用嵌套功能。

第 2 种方法是修改配置文件/etc/modprobe.d/kvm.conf 中的选项，这样可以在下次重新启动宿主机之后一直生效，示例命令如下：

```
#vi /etc/modprobe.d/kvm.conf
#Setting modprobe kvm_intel/kvm_amd nested = 1
#only enables Nested Virtualization until the next reboot or
#module reload. Uncomment the option applicable
#to your system below to enable the feature permanently.
#
#User changes in this file are preserved across upgrades.
#
#For Intel
#options kvm_intel nested=1
options kvm_intel nested=1
#
#For AMD
#options kvm_amd nested=1
```

如果是 AMD 的 CPU，则其内核模块名称为 kvm_amd，选项名为 kvm_amd nested，所以会有一些不同，示例命令如下：

```
#modprobe -r kvm_amd

#modprobe kvm_amd nested=1

#vi /etc/modprobe.d/kvm.conf
添加
options kvm_amd nested=1
```

3.3 L2 级别 KVM 宿主机的配置

实验中 L2 级别的 KVM 宿主机使用 CentOS 8.3，安装方法与普通的 KVM 虚拟化宿主机相同。

3.3.1 虚拟机配置（Intel）

安装完成后，通过 virsh edit 命令修改虚拟机的配置文件，主要编辑 CPU 的设置。示例命令如下：

```
#virsh edit centos8.3
将原有 CPU 参数
  <cpu mode='custom' match='exact' check='full'>
    <model fallback='forbid'>SandyBridge-IBRS</model>
    <vendor>Intel</vendor>
    <feature policy='require' name='vme'/>
    <feature policy='require' name='ss'/>
    <feature policy='require' name='pcid'/>
    <feature policy='require' name='hypervisor'/>
    <feature policy='require' name='arat'/>
    <feature policy='require' name='tsc_adjust'/>
    <feature policy='require' name='umip'/>
    <feature policy='require' name='md-clear'/>
    <feature policy='require' name='stibp'/>
    <feature policy='require' name='arch-capabilities'/>
    <feature policy='require' name='ssbd'/>
    <feature policy='require' name='xsaveopt'/>
    <feature policy='require' name='pdpe1gb'/>
    <feature policy='require' name='ibpb'/>
    <feature policy='require' name='amd-ssbd'/>
    <feature policy='require' name='skip-l1dfl-vmentry'/>
    <feature policy='require' name='pschange-mc-no'/>
```

```
  </cpu>
修改为
<cpu mode = 'host - passthrough'/>
```

如果 L1 级别宿主机采用的是 Intel 的 CPU，通过设置< cpu mode＝'host-passthrough'/>，则可以使 L2 宿主机像 L1 宿主机一样使用 CPU 的虚拟化特性。

也可以使用 virt-manager 设置虚拟机 CPU 的模式，如图 3-2 所示。

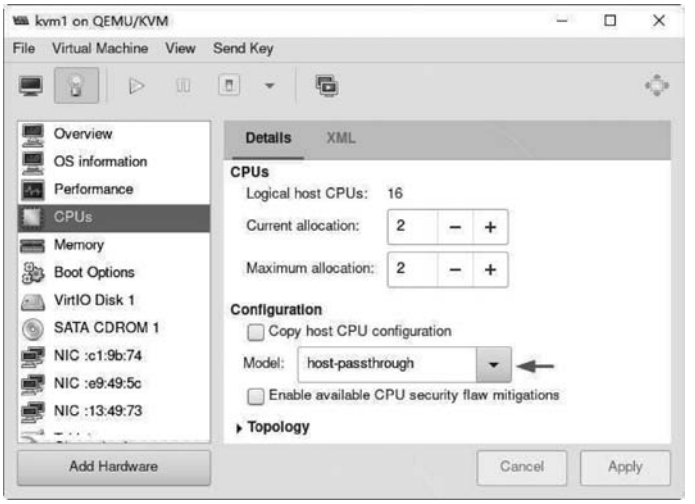

图 3-2　在 virt-manager 中设置虚拟机 CPU 的属性

目前 Cockpit 还不支持修改虚拟机的 CPU 的模式。

启动 L2 级别的虚拟机 CentOS 8.3，通过 SSH 登录，示例命令如下：

```
# virsh start centos 8.3
Domain centos8.3 started

# virsh domifaddr centos 8.3
Name       MAC address          Protocol     Address
-------------------------------------------------------------------------------
vnet0      52:54:00:ff:a7:9a    ipv4         192.168.122.146/24

# ssh 192.168.122.146
root@192.168.122.146's password:
Activate the web console with: systemctl enable --now cockpit.socket
```

L2 级别宿主机的 IP 地址是 192.168.122.146，使用 SSH 登录。通过虚拟化验证工具 virt-host-validate 进行检查，示例命令如下：

```
[root@localhost ~]# virt-host-validate
  QEMU: Checking for hardware virtualization                 : PASS
  QEMU: Checking if device /dev/kvm exists                   : PASS
  QEMU: Checking if device /dev/kvm is accessible            : PASS
  QEMU: Checking if device /dev/vhost-net exists             : PASS
  QEMU: Checking if device /dev/net/tun exists               : PASS
  QEMU: Checking for cgroup 'cpu' controller support         : PASS
  QEMU: Checking for cgroup 'cpuacct' controller support     : PASS
  QEMU: Checking for cgroup 'cpuset' controller support      : PASS
  QEMU: Checking for cgroup 'memory' controller support      : PASS
  QEMU: Checking for cgroup 'devices' controller support     : PASS
  QEMU: Checking for cgroup 'blkio' controller support       : PASS
  QEMU: Checking for device assignment IOMMU support         : WARN (No ACPI DMAR table found,
IOMMU either disabled in BIOS or not supported by this hardware platform)
  QEMU: Checking for secure guest support                    : WARN (Unknown if this platform
has Secure Guest support)
```

输出结果说明此宿主机支持虚拟化。

提示：对于嵌套虚拟化实验环境，可以忽略与 IOMMU 和 Secure Guest support 有关的警告。

查看 L2 级别宿主机的 CPU 属性，示例命令如下：

```
[root@localhost ~]# cat /proc/cpuinfo | grep model
model           : 45
model name      : Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz
model           : 45
model name      : Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz

[root@localhost ~]# cat /proc/cpuinfo | grep vmx
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon rep_good nopl
xtopology cpuid tsc_known_freq pni pclmulqdq vmx ssse3 cx16 pcid sse4_1 sse4_2 x2apic popcnt
tsc_deadline_timer aes xsave avx hypervisor lahf_lm cpuid_fault pti ssbd ibrs ibpb stibp tpr_
shadow vnmi flexpriority ept vpid tsc_adjust xsaveopt arat umip md_clear arch_capabilities
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon rep_good nopl
xtopology cpuid tsc_known_freq pni pclmulqdq vmx ssse3 cx16 pcid sse4_1 sse4_2 x2apic popcnt
tsc_deadline_timer aes xsave avx hypervisor lahf_lm cpuid_fault pti ssbd ibrs ibpb stibp tpr_
shadow vnmi flexpriority ept vpid tsc_adjust xsaveopt arat umip md_clear arch_capabilities
```

在 L2 级别的宿主机 CentOS 8.3 上，可以看到 HP Z420 工作站的 CPU 型号及特性。有了这些特性，就可以在其上再创建虚拟机了。创建的过程与直接在 L1 宿主机上创建虚拟机类似，这里不再赘述。

3.3.2　虚拟机配置(AMD)

如果 L1 级别宿主机采用的是 AMD 的 CPU,则配置 L2 级别宿主机的方法与配置 Intel 的 CPU 类似,也是将虚拟机 CPU 配置为使用 host-passthrough 模式,示例命令如下:

```
#virsh edit centos 8.3
将 CPU 属性修改为
<cpu mode = 'host - passthrough'/>
```

如果要求 L2 级别宿主机使用特定的 CPU 而不是 host-passthrough,则需在 CPU 配置中添加<feature policy='require' name='vmx'/>,示例命令如下:

```
#virsh edit centos 8.3
将 CPU 属性修改为
<cpu mode = 'custom' match = 'exact' check = 'partial'>
  <model fallback = 'allow'> Haswell - noTSX </model >
  <feature policy = 'require' name = 'vmx'/>
</cpu >
```

3.4　L2 级别 VMware ESXi 宿主机的配置

3.4.1　VMware ESXi 下载与安装

VMware vSphere Hypervisor ESXi 是一个商业产品,但是 VMware 公司提供了 60 天试用版。下面的实验使用的版本是 VMware vSphere Hypervisor(ESXi) 6.7,它的 ISO 文件的下载网址为 https://my.vmware.com/en/web/vmware/evalcenter?p=free-esxi6。

在下载页面中会显示文件的 MD5 及 SHA 类型的摘要信息,如图 3-3 所示。

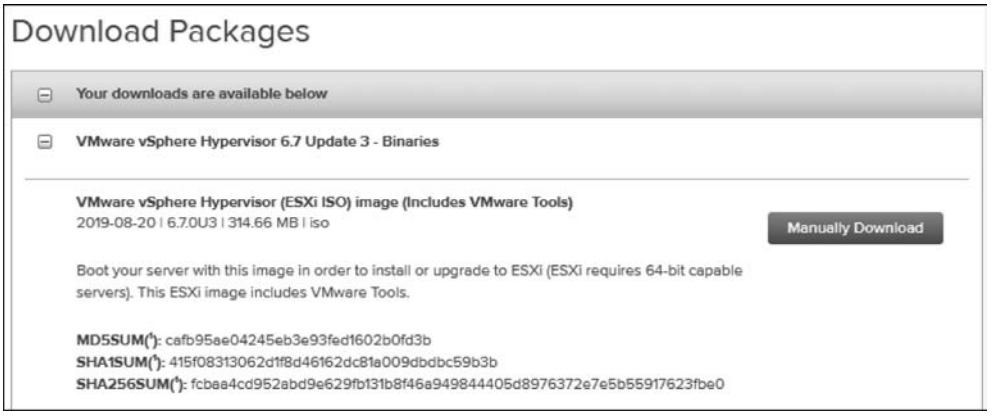

图 3-3　VMware vSphere Hypervisor 6.7 文件的信息

在安装前，建议通过这些信息来检查所下载文件的完整性，示例命令如下：

```
#md5sum /iso/VMware-VMvisor-Installer-6.7.0.update03-14320388.x86_64.iso
cafb95ae04245eb3e93fed1602b0fd3b  /iso/VMware-VMvisor-Installer-6.7.0.update03-14320388.x86_64.iso
```

由于 Cockpit 和 virt-manager 没有适合 ESXi 的操作系统参数，所以需要通过 virt-install 命令来创建能够运行 ESXi 的虚拟机，示例命令如下：

```
#virt-install --name=esxi6.7u3 \
--cpu host-passthrough \
--ram 4096 --vcpus=4 \
--virt-type=kvm --hvm \
--cdrom /iso/VMware-VMvisor-Installer-6.7.0.update03-14320388.x86_64.iso \
--network bridge=virbr1,model=e1000 \
--graphics spice,listen=127.0.0.1 \
--graphics vnc \
--video qxl \
--disk pool=vm,size=80,sparse=true,bus=ide,format=qcow2 \
--boot cdrom,hd --noautoconsole --force
```

需要注意以下选项参数。

(1) CPU：host-passthrough。

(2) 网卡类型：E1000。

(3) 磁盘接口类型：IDE。

(4) 显卡类型：QXL。

提示：ESXi 的硬件兼容列表很短，这些是经过验证的虚拟硬件类型的组合。

从 ISO 文件启动虚拟机，会出现安装的 ESXi 引导菜单，如图 3-4 所示。

图 3-4 ESXi 安装引导菜单

在欢迎屏幕上，按 Enter 键，如图 3-5 所示。

按 F11 键接受许可，如图 3-6 所示。

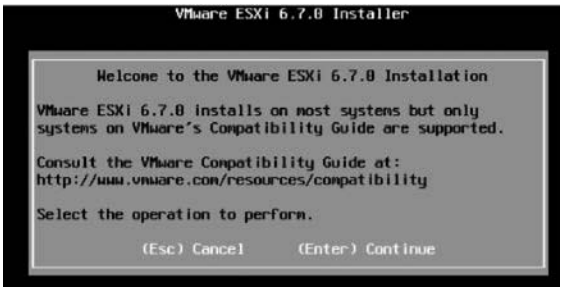

图 3-5　安装 ESXi 的欢迎信息

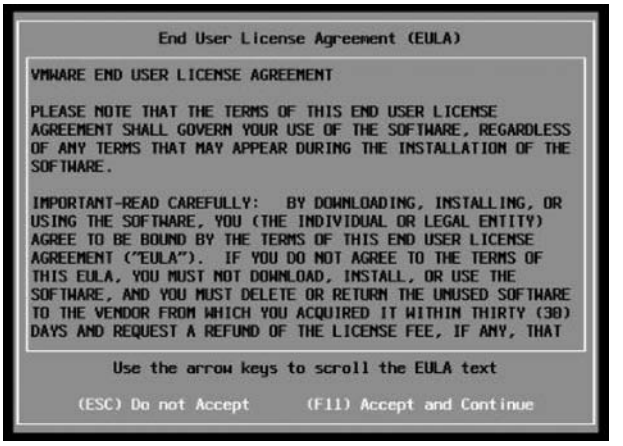

图 3-6　ESXi 的许可协议

按 Enter 键选择单个磁盘作为默认安装驱动器，如图 3-7 所示。

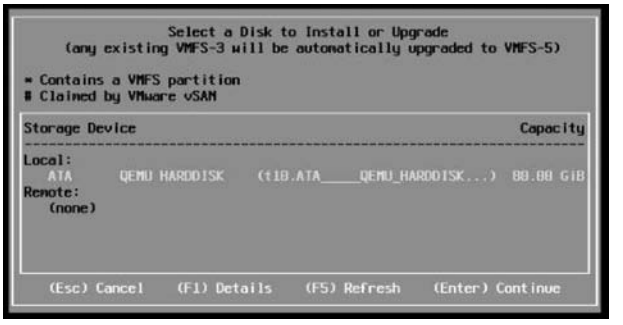

图 3-7　选择要安装或升级的磁盘

将键盘布局选为 US Default，按 Enter 键，如图 3-8 所示。

输入 root 的初始密码，如图 3-9 所示。

如果收到 CPU 或其他设备的兼容性警告，则可按 Enter 键继续，如图 3-10 所示。

图 3-8 选择键盘布局

图 3-9 设置 root 用户的密码

图 3-10 硬件兼容性警告

确认将会安装在目标磁盘,按 F11 键开始安装,如图 3-11 所示。

图 3-11 确认安装目标磁盘

安装的速度很快。在安装的过程中会显示进度条,如图 3-12 所示。

图 3-12 安装进度

安装结束后,提示断开安装介质,如图 3-13 所示。

重新启动后,会出现 ESXi 服务器的管理入口页面,如图 3-14 所示。可以使用浏览器访问屏幕上显示的地址,这是一个功能丰富的基于 Web 的管理工具。

图 3-13　安装完成

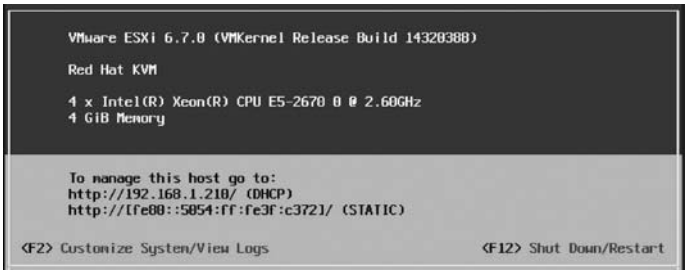

图 3-14　ESXi 服务器的管理入口页面

也可以按 F2 键，输入在安装过程中设置的 root 密码，这样就会进入系统设置界面。可以在其中完成一些最基本的设置，如图 3-15 所示。

图 3-15　系统设置界面

3.4.2　VMware ESXi 管理

通过 Web 浏览器(建议使用对 HTML5 支持比较好的浏览器，如 Chrome、Firefox 等)

打开 ESXi 的管理页面，使用 root 的用户名和密码登录，如图 3-16 所示。

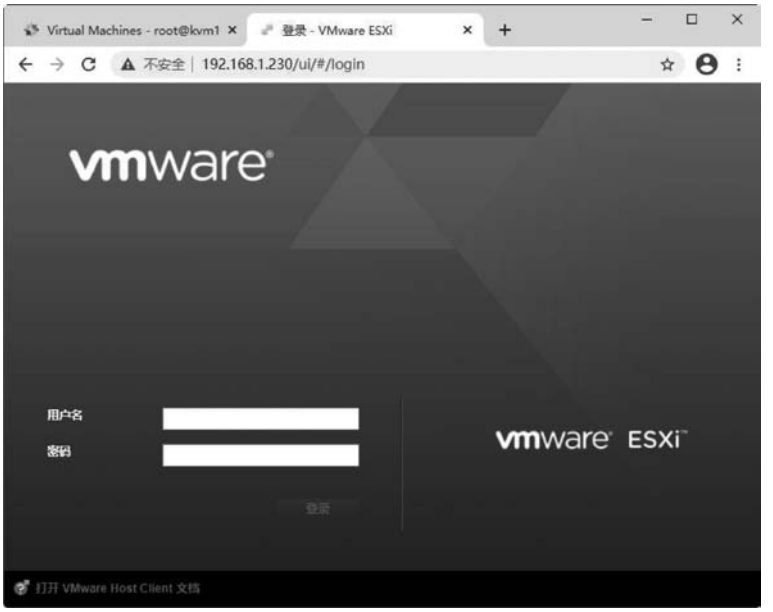

图 3-16　VMware Host Client 登录界面

这是一个 HTML/JavaScript 的应用程序，是由 ESXi 主机直接提供的轻量级管理界面，如图 3-17 所示。

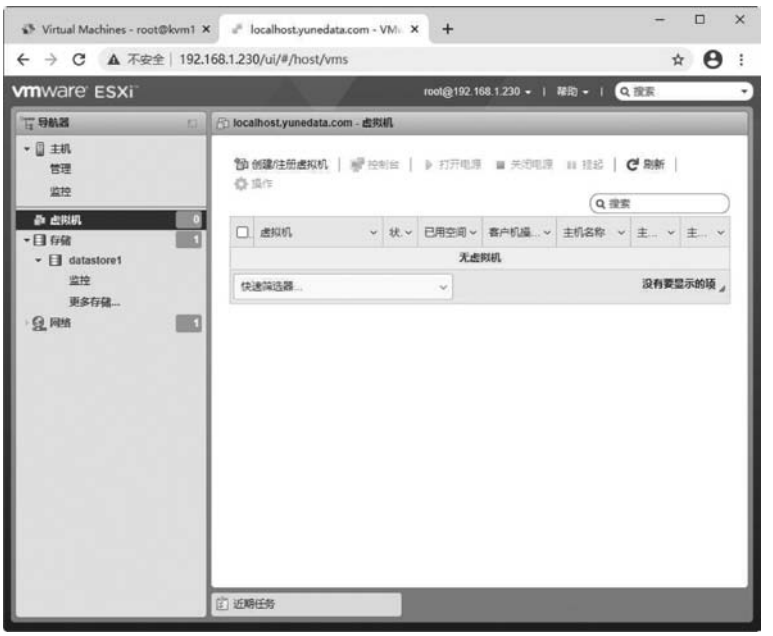

图 3-17　VMware Host Client 主界面