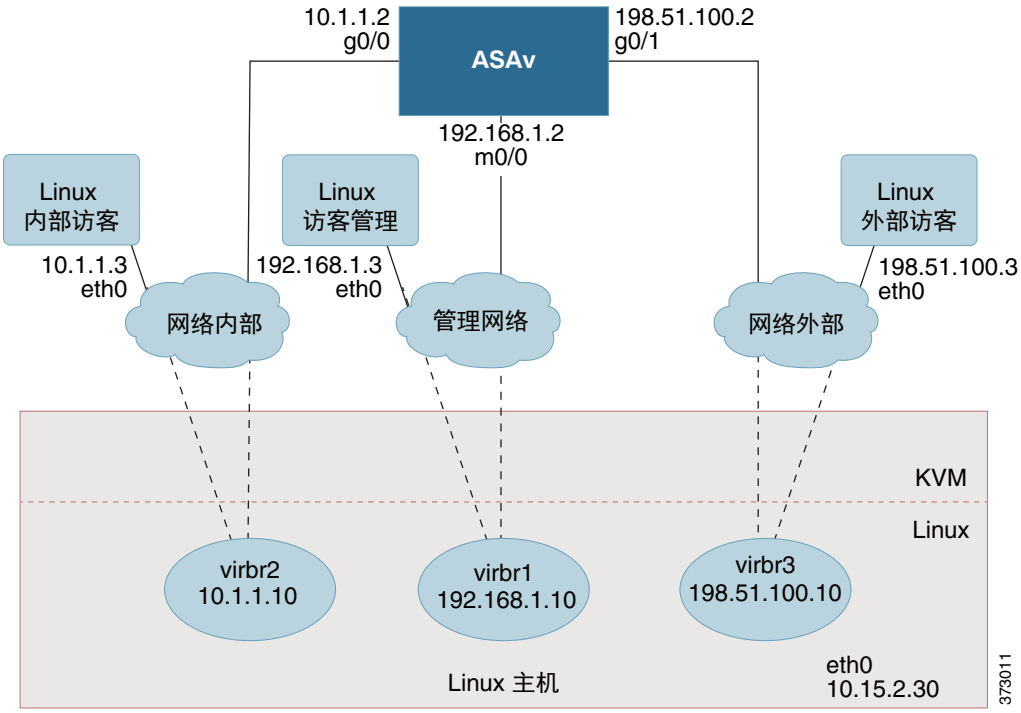
使用 KVM 部署 ASAv

您可以使用基于内核的虚拟机 (KVM) 部署 ASAv。

关于使用 KVM 的 ASAv 部署

图 1 （第 21 页）显示使用 ASAv 和 KVM 的网络拓扑示例。本章所述的程序均基于此拓扑示例。您所需的具体程序取决于您的要求。 ASAv 用作内部和外部网络之间的防火墙。另外，此示例中还配置了一个单独的管理网络。

图 1　　使用 KVM 的 ASAv 部署示例



ASAv 和 KVM 的先决条件

- 从 Cisco.com 下载 ASAv qcow2 文件并将其放在 Linux 主机上：

  http://www.cisco.com/go/asa-software

  注意：需要 Cisco.com 登录信息和思科服务合同。

- 为与本文档中的部署示例吻合，我们假定您使用 Ubuntu 14.04 LTS。将以下数据包安装在 Ubuntu 14.04 LTS 主机之上：
  - qemu-kvm
  - libvirt-bin
  - bridge-utils
  - virt-manager
  - virtinst
  - virsh tools
  - genisoimage
- 性能受主机及其配置的影响。通过调整主机，您可以最大化 KVM 上的 ASAv 吞吐量。有关通用主机调整的概念，请参阅《具备 Linux 和 Intel 架构的虚拟化平台的网络功能虚拟化数据包处理性能》。
- Ubuntu 14.04 的有用优化包括以下内容：
  - macvtap - 高性能 Linux 网桥；您可以使用 macvtap，而不是 Linux 网桥。注意，您必须配置特定设置才能使用 macvtap，而不是 Linux 网桥。
  - 透明大页面 (Transparent Huge Pages) - 用于增加内存页面大小，在 Ubuntu 14.04 中默认开启。
  - 禁用超线程 (Hyperthread disabled) - 用于将两个 vCPU 减少到一个单核。
  - txqueuelength - 用于将默认 txqueuelength 增加到 4000 个数据包并减少丢包率。
  - 固定 (pinning) - 用于将 qemu 和 vhost 进程固定到特定 CPU 内核；在某些情况下，固定可显著提高性能。
- 有关优化基于 RHEL 的分发的信息，请参阅 《Red Hat Enterprise Linux6 虚拟化调整和优化指南》。
- 有关 KVM 的系统要求，请参阅思科 ASA 兼容性矩阵。

准备 Day 0 配置文件

在启动 ASAv 之前，您可以准备 Day 0 配置文件。此文件是包含将在 ASAv 启动时应用的 ASAv 配置的文本文件。此初始配置将放入您选择的工作目录中名为“day0-config”的文本文件，并写入首次启动时安装和读取的 day0.iso 文件。 Day 0 配置文件必须至少包含将激活管理接口以及设置用于公钥身份验证的 SSH 服务器的命令，但它还可包含完整的 ASA 配置。day0.iso 文件 （自定义 day0.iso 或默认 day0.iso）必须在首次启动过程中可用。

注意：要在初始部署过程中自动授权 ASAv，请将从思科智能软件管理器下载的智能许可身份 (ID) 令牌放入与 Day 0 配置文件处于同一目录且名为“idtoken”的文本文件。

注意：如果要在透明模式下部署 ASAv，则必须在透明模式下将已知的运行 ASA 配置文件用作 Day 0 配置文件。这不适用于路由防火墙的 Day 0 配置文件。

注意：我们在本示例中使用的是 Linux，但对于 Windows 也有类似的实用程序。

程序

1. 在名为“day0-config”的文本文件中输入 ASAv 的 CLI 配置。添加三个接口的接口配置和所需的任何其他配置。

   第一行应以 ASA 版本开头。day0-config 应该是有效的 ASA 配置。生成 day0-config 的最佳方式是从现有的 ASA 或 ASAv 复制一个运行配置的所需部分。 day0-config 中的行顺序很重要，应与现有的 **show run** 命令输出中看到的顺序相符。

示例：

```
ASA Version 9.5.1
!
interface management0/0
   nameif management
   security-level 100
   ip address 192.168.1.2 255.255.255.0
   no shutdown
interface gigabitethernet0/0
   nameif inside
   security-level 100
   ip address 10.1.1.2 255.255.255.0
   no shutdown
interface gigabitethernet0/1
   nameif outside
   security-level 0
   ip address 198.51.100.2 255.255.255.0
   no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
```

2. （可选）将思科智能软件管理器发布的智能许可证身份令牌文件下载到您的计算机。
3. （可选）从下载文件复制 ID 令牌并将其放入仅包含 ID 令牌的、名为“idtoken”的文本文件。
4. （可选）若要在初始 ASAv 部署过程中进行自动许可，请确保 day0-config 文件中包含以下信息：
   - 管理接口 IP 地址
   - （可选）要用于智能许可的 HTTP 代理
   - 用于启用与 HTTP 代理 （如果指定）或 tools.cisco.com 的连接的 **route** 命令
   - 将 tools.cisco.com 解析为 IP 地址的 DNS 服务器
   - 指定您正请求的 ASAv 许可证的智能许可配置
   - （可选）更加便于 ASAv 在 CSSM 中进行查找的唯一主机名
5. 通过将文本文件转换成 ISO 文件生成虚拟 CD-ROM：

```
stack@user-ubuntu:-/KvmAsa$ sudo genisoimage -r -o day0.iso day0-config idtoken
I: input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 252
Total directory bytes: 0
Path table size (byptes): 10
Max brk space used 0
176 extents written (0 MB)
stack@user-ubuntu:-/KvmAsa$
```

   身份令牌自动向智能许可服务器注册 ASAv。

6. 重复步骤 1 到 5，使用相应的 IP 地址为要部署的每个 ASAv 创建单独的默认配置文件。

准备虚拟网桥 XML 文件

您需要设置将 ASAv 访客连接到 KVM 主机，以及将访客彼此连接的虚拟网络。

注意：此程序不会建立与 KVM 主机之外的外部环境的连接。

在 KVM 主机上准备虚拟网桥 XML 文件。对于准备 Day 0 配置文件 （第 22 页）所述的虚拟网络拓扑示例，您需要以下三个虚拟网桥文件：virbr1.xml、 virbr2.xml 和 virbr3.xml （您必须使用这三个文件名；例如，不允许使用 virbr0，因为它已经存在）。每个文件具有设置虚拟网桥所需的信息。您必须为虚拟网桥提供名称和唯一的 MAC 地址。提供 IP 地址是可选的。

程序

1. 创建三个虚拟网络网桥 XML 文件：

   virbr1.xml：

   ```
   <network>
     <name>virbr1</name>
     <bridge name='virbr1' stp='on' delay='0' />
     <mac address='52:54:00:05:6e:00' />
     <ip address='192.168.1.10' netmask='255.255.255.0' />
   </network>
   ```

   virbr2.xml：

   ```
   <network>
     <name>virbr2</name>
     <bridge name='virbr2' stp='on' delay='0' />
     <mac address='52:54:00:05:6e:01' />
     <ip address='10.1.1.10' netmask='255.255.255.0' />
   </network>
   ```

   virbr3.xml：

   ```
   <network>
     <name>virbr3</name>
     <bridge name='virbr3' stp='on' delay='0' />
     <mac address='52:54:00:05:6e:02' />
     <ip address='198.51.100.10' netmask='255.255.255.0' />
   </network>
   ```

2. 创建包含以下内容的脚本 （在本例中，我们将脚本命名为 virt_network_setup.sh）：

   ```
   virsh net-create virbr1.xml
   virsh net-create virbr2.xml
   virsh net-create virbr3.xml
   ```

3. 运行此脚本以设置虚拟网络。此脚本将生成虚拟网络。只要 KVM 主机运行，网络就会保持运行。

   ```
   stack@user-ubuntu:-/KvmAsa$ virt_network_setup.sh
   ```

   注意：如果重新加载 Linux 主机，则必须重新运行 virt_network_setup.sh 脚本。此脚本在主机重启期间即停止运行。

4. 验证虚拟网络是否已创建：

   ```
   stack@user-ubuntu:-/KvmAsa$ brctl show
   bridge name     bridge id               STP enabled     Interfaces
   virbr0          8000.0000000000000      yes
   virbr1          8000.5254000056eed      yes             virb1-nic
   virbr2          8000.5254000056eee      yes             virb2-nic
   virbr3          8000.5254000056eec      yes             virb3-nic
   stack@user-ubuntu:-/KvmAsa$
   ```

5. 显示分配给 virbr1 网桥的 IP 地址。这是您在 XML 文件中分配的 IP 地址。

```
stack@user-ubuntu:-/KvmAsa$ ip address show virbr1
S: virbr1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 52:54:00:05:6e:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 brd 192.168.1.255 scope global virbr1
      valid_lft forever preferred_lft forever
```

启动 ASAv

使用基于 virt-install 的部署脚本启动 ASAv。

程序

1. 创建名为“virt_install_asav.sh”的 virt-install 脚本。

   ASAv VM 的名称在此 KVM 主机上的所有其他虚拟机 (VM) 中必须是唯一的。 ASAv 最多可以支持 10 个网络。此示例使用三个网络。网络网桥语句的顺序非常重要。第一个列出的始终是 ASAv 的管理接口 (Management 0/0)，第二个列出的是 ASAv 的 GigabitEthernet 0/0，第三个列出的是 ASAv 的 GigabitEthernet 0/1，以此类推，直至 GigabitEthernet0/8。虚拟 NIC 必须是 Virtio。

   注意：*watchdog* 要素是 KVM 访客的虚拟硬件监视设备。如果 ASAv 因任何原因而变得无响应，监视设备可以触发重新启动 KVM 访客。

```
virt-install \
   --connect=qemu:///system \
   --network network=default,model=virtio \
   --network network=default,model=virtio \
   --network network=default,model=virtio \
   --name=asav \
   --cpu host \
   --arch=x86_64 \
   --machine=pc-1.0 \
   --vcpus=1 \
   --ram=2048 \
   --os-type=linux \
   --os-variant=generic26 \
   --noacpi \
   --virt-type=kvm \
   --import \
    --watchdog i6300esb,action=reset
   --disk path=/home/kvmperf/Images/desmo.qcow2,format=qcow2,device=disk,bus=ide,cache=none \
   --disk path=/home/kvmperf/asav_day0.iso,format=iso,device=cdrom \
   --console pty,target_type=virtio \
   --serial tcp,host=127.0.0.1:4554,mode=bind,protocol=telnet
```

2. 运行 virt_install 脚本：

```
stack@user-ubuntu:-/KvmAsa$ ./virt_install_asav.sh

Starting install...
Creating domain...
```

   系统将显示窗口，其中显示 VM 的控制台。您可以看到 VM 正在启动。 VM 需要几分钟进行启动。 VM 停止启动后，您可以从控制台屏幕发出 CLI 命令。

热插拔接口调配

您可以动态添加和删除接口，而无需停止并重新启动 ASAv。在将新的接口添加到 ASAv 虚拟机时， ASAv 应该能够检测到该接口，并且将其调配为常规接口。同样，当您通过热插拔调配的方式删除现有的接口时， ASAv 应删除该接口并释放与其相关联的任何资源。

热插拔接口调配的准则

接口映射与编号

- 当您添加一个热插拔接口时，其接口编号等于当前的最后一个接口的编号加上 1。
- 当您删除一个热插拔接口时，会产生一个接口编号缺口，除非您删除的接口是最后一个接口。
- 当存在一个接口编号缺口时，下一个热插拔调配的接口将填补该缺口。

故障切换

- 在将热插拔接口用作故障切换链路时，必须在指定为故障切换 ASAv 对的两台设备上调配该链路。
  - 首先将一个热插拔接口添加到虚拟机监控程序中的主用 ASAv，然后将一个热插拔接口添加到虚拟机监控程序中的备用 ASAv。
  - 在主用 ASAv 中配置新添加的故障切换接口；该配置将同步到备用设备。
  - 在主设备上启用故障切换。
- 要删除故障切换链路，请执行以下操作：
  - 首先删除主用 ASAv 中的故障切换配置。
  - 从虚拟机监控程序内的主用 ASAv 中删除故障切换接口，然后立即从虚拟机监控程序内的备用 ASAv 中删除相应的接口。

限制

- 热插拔接口调配限于 Virtio 虚拟 NIC。
- 支持的最大接口数量是 10。如果您尝试添加超过 10 个接口，则会收到错误消息。
- 您无法打开接口卡 (media_ethernet/port/id/10)。

您可以使用 virsh 命令行添加和删除 KVM 虚拟机监控程序中的接口。

程序

1. 打开 virsh 命令行会话：

```
[root@asav-kvmterm ~]# virsh
Welcome to virsh, the virtualization interactive terminal.

Type:    'help' for help with commands
         'quit' to quit
```

2. 使用 **attach-interface** 命令添加一个接口：

```
virsh # attach-interface domain type source model mac live
```

示例：

```
virsh # attach-interface --domain asav-network --type bridge --source br_hpi --model virtio --mac
52:55:04:4b:59:2f --live
```

Domain 可以指定为短整数、名称或完整的 UUID。 *type* 参数可以是“network” （表示物理网络设备）或“bridge” （表示连接到设备的网桥）。*source* 参数表示连接类型。*model* 参数表示虚拟 NIC 类型。*mac* 参数指定网络接口的 MAC 地址。*live* 参数表示该命令影响正在运行的域。

注意：请使用 ASAv 上的接口配置模式配置并启用该接口，以便传输和接收流量；有关详细信息，请参阅思科 ASA 系列文档导航。

3. 使用 **detach-interface** 命令删除一个接口：

```
virsh # detach-interface domain type mac live
```

示例：

```
virsh # detach-interface --domain asav-network --type bridge --mac 52:55:04:4b:59:2f --live
```

热插拔接口调配