

Virtualização com KVM no Debian Jessie

De Eriberto Wiki

Índice

- 1 Introdução: sai Xen, entra KVM
- 2 Instalação inicial
- 3 Criação de uma máquina virtual Debian
- 4 Possíveis problemas na primeira execução do script
 - 4.1 KVM kernel module: Permission denied
 - 4.2 Cancelamento da instalação para o reinício da mesma

Introdução: sai Xen, entra KVM

Por muito tempo eu usei Xen. Hoje, só aconselho KVM + LibVirt. Os motivos são os seguintes:

- As máquinas evoluíram e, hoje em dia, possuem recursos de hardware para virtualização, acelerando processos.
- Xen é nativo no kernel mas é especialista em paravirtualização.
 - A paravirtualização era um importante recurso na década de 2000, pois exigia pouco hardware. Hoje o hardware é muito mais potente e entende bem virtualização completa.
 - Na para virtualização só é possível virtualizar o mesmo sistema do hospedeiro.
 - A atualização de SO na paravirtualização é extremamente complexa e, geralmente, exige a parada total do sistema.
 - O Xen demanda em mais de 500.000 linhas de código.
- KVM também é nativo do kernel e é especialista em virtualização completa.
 - A virtualização completa permite qualquer SO sobre qualquer SO.
 - Na virtualização completa as atualizações de software são fáceis e transparentes.
 - O KVM demanda em, apenas, um pouco mais de 10.000 linhas de código.
 - Mesmo realizando virtualização completa, o KVM demonstrou, em vários testes, ter performance superior ao Xen.
- A LibVirt é uma API específica para virtualização. Ela provê uma linguagem única para a gerência de vários virtualizadores, além de drivers específicos para a melhoria de performance.

Para dados mais consistentes e testes comparativos, leia os seguintes documentos:

- Common considerations when selecting your hypervisor (<https://www.flexiant.com/2014/02/12/hypervisor-comparison-kvm-xen-vmware-hyper-v>), da Flexiant. É necessário se cadastrar para fazer o download. O artigo compara KVM, Xen, VMware e Hyper-V.
- Performance benchmarks: KVM vs. Xen (<https://major.io/2014/06/22/performance-benchmarks-kvm-vs-xe>), escrito por Major Hayden.
- Performance Comparison of KVM, VMware and XenServer using a Large Telecommunication Application ([http://www.bth.se/people/ssb.nsf/attachments/Performance%20Comparison%20of%20KVM,%20VMware%20and%20XenServer%20using%20a%20Large%20Telecommunication%20Application_fullpaper_Camera-ready_pdf/\\$file/Performance%20Comparison%20of%20KVM,%20VMware%20and%20XenServer%20using%20a%20Large%20Telecommunication%20Application_fullpaper_Camera-ready.pdf](http://www.bth.se/people/ssb.nsf/attachments/Performance%20Comparison%20of%20KVM,%20VMware%20and%20XenServer%20using%20a%20Large%20Telecommunication%20Application_fullpaper_Camera-ready_pdf/$file/Performance%20Comparison%20of%20KVM,%20VMware%20and%20XenServer%20using%20a%20Large%20Telecommunication%20Application_fullpaper_Camera-ready.pdf))

Para uma visão geral sobre a LibVirt, leia a página principal do projeto em <http://libvirt.org>.

Instalação inicial

Na máquina real (que será a hospedeira), instale os pacotes *kvm*, *libvirt-bin*, *virtinst* e *virt-top*.

```
# apt-get install kvm libvirt-bin virtinst virt-top
```

O próximo passo será prover uma interface de rede, como bridge, para que as máquinas virtuais possam utilizar. A máquina real também poderá usar essa mesma interface. Partindo desse princípio, vamos a duas possibilidades de configuração: com IP fixo e com DHCP.

Para a primeira possibilidade, edite o arquivo */etc/network/interfaces* e deixe a configuração parecida com a mostrada a seguir, atentando para as partes em negrito:

```
auto br-kvm
iface br-kvm inet static
    address 192.168.10.10
    netmask 255.255.255.0
    gateway 192.168.10.1
    bridge_ports eth0
```

Note que a configuração original da placa de rede, antes da bridge, era a seguinte:

```
auto eth0
iface eth0 inet static
    address 192.168.10.10
    netmask 255.255.255.0
    gateway 192.168.10.1
```

Agora, a configuração referente à *eth0* não deverá mais existir (exclua qualquer bloco de configuração referente à *eth0*).

Depois de realizar a configuração, execute os comandos:

```
# ifconfig eth0 0.0.0.0
# /etc/init.d/networking restart
```

O comando *ifconfig* deverá mostrar:

```
br-kvm  Link encap:Ethernet  Endereço de HW 00:e0:4c:50:44:58
        inet end.: 192.168.10.10  Bcast:192.168.10.255  Masc:255.255.255.0
        UP BROADCASTMULTICAST  MTU:1500  Métrica:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth0    Link encap:Ethernet  Endereço de HW 00:e0:4c:50:44:58
        UP BROADCASTMULTICAST  MTU:1500  Métrica:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Note que a interface *eth0* não possui endereço IP. A máquina real funcionará e poderá ser acessada normalmente pela interface *br-kvm*.

Caso a interface *br-kvm* não seja mostrada corretamente, tente resolver o problema com os comandos *ifup* e *ifdown*. Se você não tiver habilidade com esses comandos, reinicie a máquina e tudo deverá funcionar.

Para o caso de DHCP, a configuração ficará da seguinte forma:

```
auto br-kvm
iface br-kvm inet dhcp
```

```
bridge_ports eth0
```

Caso você precise de várias interfaces para as máquinas virtuais, você poderá criar várias bridges, como *br-kvm1*, *br-kvm2* etc., desde que haja interfaces reais, como *eth1*, *eth2* etc.

Criação de uma máquina virtual Debian

As máquinas virtuais poderão ser criadas facilmente com o comando *virt-install*. Assim sendo, crie um script shell, denominado *cria-vm.sh*, com o seguinte conteúdo:

```
#!/bin/bash
# Alterar, de acordo com a máquina a ser criada
#
NOME=teste
VCPUS=2
RAM=4096
MAC=52:54:00:00:00:01
# Não alterar a partir daqui
virt-install -d \
  --name=$NOME \
  --vcpus=$VCPUS \
  --ram $RAM \
  --disk path=/vms/$NOME,bus=virtio,cache=none \
  --network bridge=br-kvm,model=virtio,mac=$MAC \
  --accelerate \
  --graphics none \
  --extra-args="console=ttyS0,115200" \
  --location=http://ftp.debian.org/debian/dists/jessie/main/installer-amd64
```

No primeiro bloco, altere, de acordo com a necessidade, o nome da máquina (NOME), a quantidade de núcleos (VCPUS), a quantidade de memória RAM em gigabytes (RAM) e o endereço MAC a ser utilizado pela máquina (MAC). No caso do MAC, o range 52:54:00:xx:xx:xx é destinado à LibVirt. Assim, é interessante usar sempre esse range.

A seguir, crie um diretório */vms* (poderá ser em outro local, diferente da raiz, se necessário).

```
# mkdir /vms
```

Dentro do */vms*, crie um arquivo para receber a máquina virtual (sim, a máquina ficará dentro de um arquivo). Esse arquivo deverá ter o mesmo nome da máquina a ser criada. Para um teste rápido ou para um servidor pequeno, como um DNS ou DHCP, poderá ser utilizado um arquivo de 10 GB. Use o comando *dd* para criar o arquivo.

```
# dd if=/dev/zero of=/vms/teste bs=1G count=10
```

A linha *--disk* do script define que o arquivo será buscado em */vms*.

Uma outra definição importante, provida pela linha *--network*, refere-se à placa de rede da máquina real (em modo bridge) que será utilizada para o tráfego.

Ao executar o script, a máquina será criada. Não esqueça de prover permissão de execução antes de rodar tal script:

```
# chmod 750 cria-vm.sh
# ./cria-vm.sh
```

Ao rodar o script, o mesmo baixará dois pequenos arquivos de instalação do Jessie amd64. Veja que isso é definido na linha *--location* do script e pode ser mudado.

Possíveis problemas na primeira execução do script

Alguns problemas poderão ocorrer durante a primeira execução do script *cria-vm.sh*. Veremos algumas possibilidades.

KVM kernel module: Permission denied

Ao executar o script *cria-vm.sh* pela primeira vez, poderá ocorrer a seguinte erro:

```
Internal error: process exited while connecting to monitor: Could not access KVM kernel module: Permission denied
```

Isso ocorre porque o KVM gera um dispositivo em */dev* (*/dev/kvm*) e carrega dois módulos de kernel (*kvm* e *kvm_intel*). O problema é que as permissões de acesso ao */dev/kvm* são setadas, pela primeira e única vez, depois que os módulos são carregados. Assim sendo, há impacto na utilização do dispositivo. Há duas formas de resolver isso:

- A mais simples: reiniciando a máquina real.
- A mais técnica: recarregando os módulos.

Para a segunda solução, utilize os comandos:

```
# modprobe -r kvm_intel
# modprobe -r kvm
# modprobe kvm_intel
```

Cancelamento da instalação para o reinício da mesma

É possível que a instalação seja cancelada por dois motivos:

- Erro ao executar o script.
- Mediante um Ctrl-C por vontade do usuário.

Nesses dois casos, teremos uma instância da máquina rodando em background e a associação da mesma ao sistema. Com isso, a execução do script *cria-vm.sh* novamente irá gerar o seguinte erro:

```
Guest name 'teste' is already in use.
```

Para executar o script após um break, será necessário, antes de tudo, emitir os seguintes comandos:

```
# virsh destroy teste
# virsh undefine teste
```

Disponível em "http://eriberto.pro.br/wiki/index.php?title=Virtualização_com_KVM_no_Debian_Jessie&oldid=2354"

- Esta página foi modificada pela última vez à(s) 07h31min de 11 de dezembro de 2015.
- Esta página foi acessada 607 vezes.
- Conteúdo disponível sob Creative Commons - Atribuição - Uso Não Comercial - Partilha nos Mesmos Termos, salvo indicação em contrário.