

Virtualisation avec KVM

Déterminer la configuration système appropriée pour votre environnement

Il s'agit d'une étape très importante et nous devons réussir. Avoir la bonne configuration système est la clé pour obtenir des performances natives des machines virtuelles.

Commençons par le CPU.

Physical CPU

Un processeur Intel ou AMD 64 bits doté d'une extension de virtualisation, VT-x pour Intel et AMD-V pour AMD.

Pour déterminer si votre CPU prend en charge l'extension de virtualisation, vous pouvez vérifier les éléments suivants (flags):

```
# grep --color -Ew 'svm|vmx|lm' /proc/cpuinfo
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 rdtscp lm
constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc
aperfperf pni dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm
sse4_1 sse4_2 popcnt lahf_lm ida dtherm tpr_shadow vnmi flexpriority ept vpid
```

The `svm` flag means that the CPU has AMD-V, `vmx` flag means that the CPU has VT-x, and `lm` means a 64-bit support.

Si votre CPU prend en charge une extension de virtualisation, votre système est probablement prêt à héberger les machines virtuelles KVM. Vous remarquerez également que les modules KVM appropriés sont chargés automatiquement sans configuration supplémentaire. Pour vérifier si les modules sont chargés ou non, utilisez la commande suivante:

```
# lsmod | grep kvm
```

Si le système est AMD, vous verrez `kvm_amd` au lieu de `kvm_intel`.

Si vous ne voyez pas les drapeaux CPU précédents ou que tous les modules KVM ne sont pas chargés, mais que vous êtes sûr que le système prend en charge les extensions de virtualisation, essayez les étapes de dépannage suivantes:

1. Redémarrez le système et accédez au BIOS.
2. Accédez aux options avancées du processeur. Activez la technologie de virtualisation Intel ou les extensions de virtualisation. Pour AMD, il doit être activé par défaut. Les mots exacts peuvent être différents selon votre BIOS.
3. Redémarrez la machine.
4. Vous devriez maintenant voir les modules KVM chargés. Si vous ne les voyez toujours pas comme chargés, essayez de les charger manuellement.

```
# modprobe kvm kvm_intel or modprobe kvm kvm_amd
```

If you are able to load them manually but they still don't work, then it is time to involve your hardware vendor or double-check the processor details on respective Intel or AMD product pages.

En plus de l'extension de virtualisation, vous devrez peut-être activer Intel VT-d ou AMD IOMMU (AMD-Vi) dans le BIOS. Celles-ci sont requises pour l'attribution directe de périphériques PCI aux machines virtuelles, par exemple pour affecter une carte d'interface réseau (NIC) physique de l'hyperviseur à la machine virtuelle; nous en parlerons davantage dans les prochains chapitres.

1. `#egrep '(vmx|svm)' /proc/cpuinfo`

```
# egrep -c '(vmx|svm)' /proc/cpuinfo
```

If **0** it means that your CPU doesn't support hardware virtualization.
If **1** or more it does - but you still need to make sure that virtualization is enabled in the BIOS.

2. Que remarquez-vous?
3. Taper **#kvm-ok** (**après avoir installé cpu-checker**)
4. Ajuster au niveau du Bios et redémarrer
5. Remarque, Si vous voyez :

INFO: Your CPU does not support KVM extensions KVM acceleration can NOT be used

You can still run virtual machines, but it'll be much slower without the KVM extensions.

NOTE: You may see a message like "KVM acceleration can/can NOT be used". This is misleading and only means if KVM is *currently* available (i.e. "turned on"), *not* if it is supported.

6. Vérifier en tapant encore une fois **kvm-ok**
7. On recherche la version du noyau en cours d'exécution **#uname -a**
8. Quelle est la version ?
9. Installation des paquets:

```
#aptitude install libvirt-bin kvm qemu virtinst qemu-kvm virt-  
manager virt-viewer bridge-utils ubuntu-vmbuilder
```

10. Vérifier le cpu (64 ou 32 ?)

Use a 64 bit kernel (if possible)

Running a 64 bit kernel on the host operating system is recommended but not required.

- To serve more than 2GB of RAM for your VMs, you **must** use a 64-bit kernel
- On a 32-bit kernel install, you'll be limited to 2GB RAM at maximum for a given VM.
- Also, a 64-bit system can host both 32-bit and 64-bit guests.
- A 32-bit system can only host 32-bit guests.

Pour verifier si votre processor est 64-bit,

```
egrep -c ' lm ' /proc/cpuinfo
```

If 0 is printed, it means that your CPU is not 64-bit.

If 1 or higher, it is. Note: *lm* stands for Long Mode which equates to a 64-bit CPU.

Now see if your running kernel is 64-bit, just issue the following command:

```
uname -m
```

`x86_64` indicates a running 64-bit kernel. If you use see `i386`, `i486`, `i586` or `i686`, you're running a 32-bit kernel.

Note: `x86_64` is synonymous with `amd64`.

11. Si les extensions de virtualisation sont bien gérées par le processeur, le script d'init de kvm a dû charger automatiquement les modules noyaux. Il est possible de le vérifier avec la commande suivante :

```
#lsmod | grep kvm
```

12. Vérification de l'installation

You can test if your install has been successful with the following command:

```
$ virsh -c qemu:///system list
Id      Name      State
-----
$
```

If on the other hand you get something like this:

```
$ virsh -c qemu:///system list
libvir: Remote error : Permission denied error: failed to connect to
the hypervisor $
```

Something is wrong (e.g. you did not relogin) and you probably want to fix this before you move on. The critical point here is whether or not you have write access to `/var/run/libvirt/libvirt-sock`.