

Dernière mise-à-jour : 2021/01/24 12:16

HAR500 - Gestion de la Virtualisation avec Xen

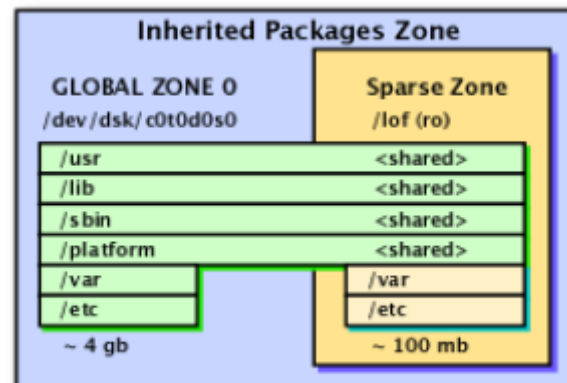
Présentation

Le système de base hébergeant les machines virtuelles est appelé l'**hôte** tandis que les machines virtuelles sont appelées les **invités**.

Il existe différentes méthodes de virtualisation :

- **Virtualisation au niveau du système d'exploitation ou *L'Isolation***

- **Description** : Les systèmes invités utilisent le même noyau et une partie du système de fichiers de l'hôte.
- **L'avantage principal** : Économise des ressources.
- **L'inconvénient principal** : Les OS des invités ont besoin d'être modifiés pour fonctionner en tant que machine virtuelle. L'hôte et les invités doivent utiliser le même OS.
- **Logiciels concernés** : OpenVZ, Linux VServer, BSD Jails, LXC (Linux Containers), Solaris Zones.

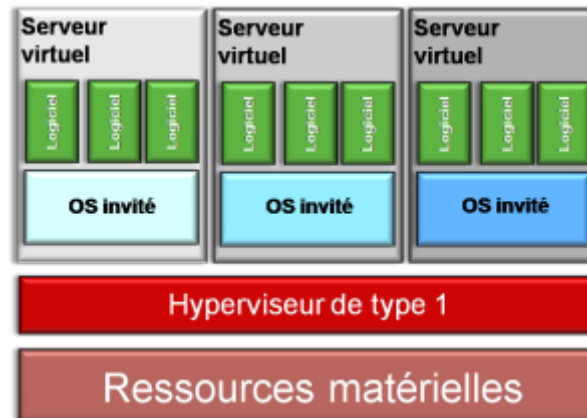


- **Paravirtualisation ou Hyperviseur de type 1**

- **Description** : Le système hôte met à disposition une machine virtuelle dans laquelle les autres invités s'exécutent. Les systèmes invités

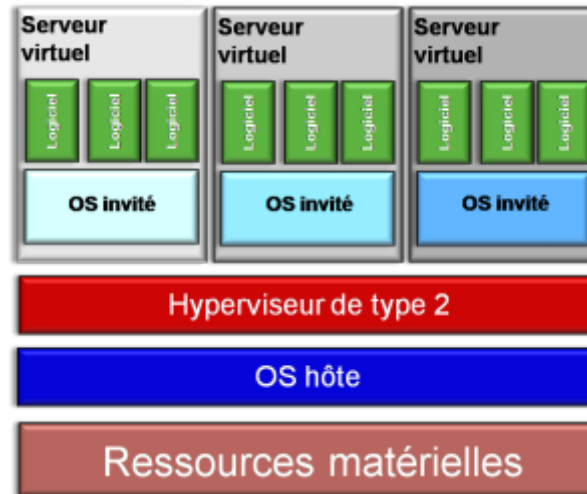
sont modifiés et communiquent directement avec l'hyperviseur.

- **L'avantage principal** : Efficacité.
- **L'inconvénient principal** : Les OS des invités ont besoin d'être modifiés pour fonctionner en tant que machine virtuelle.
- **Logiciels concernés** : Xen, VMWare™ ESX, Microsoft™ Hyper-V.



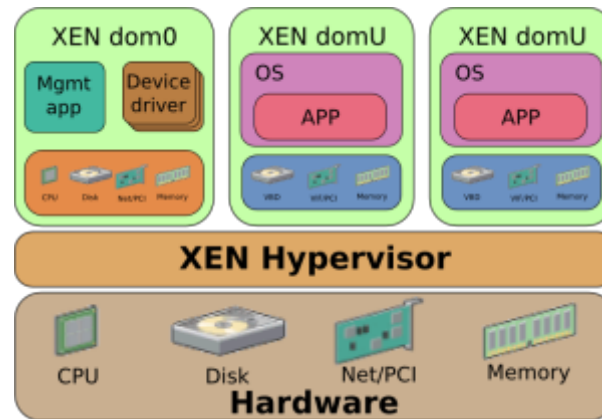
- **Virtualisation complète ou Hyperviseur de type 2**

- **Description** : Le programme simule du matériel virtuel qui apparaît du point de vue de l'invité comme du matériel réellement existant. Un **hyperviseur** ou **VMM** (*Virtual Machine Manager*) contrôle l'invité et remplace certaines opérations par d'autres afin de gérer le processeur, le disque dur, la mémoire, les processus etc..
- **L'avantage principal** : Les OS des invités n'ont pas besoin d'être modifiés pour fonctionner en tant que machine virtuelle.
- **L'inconvénient principal** : La lenteur.
- **Logiciels concernés** : VMWare™ Fusion, VMWare™ Player, VMWare™ Server, VMWare™ Fusion, Parallels Desktop, Parallels Server, Sun/Oracle VirtualBox, Microsoft™ VirtualPC, Microsoft™ VirtualServer, QEMU, BOSCH.



- **Paravirtualisation avec prise en charge de matériel**

- **Description** : Les processeurs Intel-VT et AMD-V contiennent des instructions matérielles pour faciliter la virtualisation. Pour déterminer si le processeur dispose des fonctionnalités de virtualisation matérielles, soit Intel-VT, soit AMD-V, lancez la commande **# egrep '^flags.*(vmx|svm)' /proc/cpuinfo [Entrée]**. Dans le cas où vous ne voyez rien, le processeur ne dispose **pas** de fonctionnalités de virtualisation matérielles. Par contre, même dans le cas de la prise en charge, vérifiez que les fonctionnalités soient activées dans le BIOS de la machine.
- **L'avantage principal** : Efficacité liée au fait que la plupart des OS des invités n'ont pas besoin d'être modifiés pour fonctionner en tant que machine virtuelle.
- **L'inconvénient principal** : Nécessite un processeur spécial.
- **Logiciels concernés** : Xen, KVM



Xen

- **Xen** a vu le jour en 2001 à l'**Université de Cambridge**,
- Xen est un produit en licence GPL,
- Il existe des systèmes de virtualisation commerciaux à base de Xen dont le plus connu est actuellement **Citrix XenServer**,
- Xen est un système de virtualisation principalement destiné à la virtualisation de serveurs,
- Xen est un système de **paravirtualisation** qui nécessite un noyau Linux modifié,
- Xen ne peut pas lancer donc un système non-modifié tel Windows™ en mode paravirtualisation,
- Xen peut lancer des systèmes non modifiés dans des **HVM** (*Hardware Virtual Machine*) depuis sa version 3 en utilisant une partie du code de QEMU combinée avec l'utilisation d'un processeur de virtualisation

Architecture et terminologie

- Les fonctions de base de Xen sont exécutées dans une couche du noyau,
- Xen ne démarre que le premier invité,
- Lors du démarrage de la machine :
 - Xen est démarré en tant qu'hôte,
 - L'hôte des invités est démarré en tant que **Domain0** (Dom0), un domain dit **privilegié**,
 - Le dom0 démarre les domaines **non-privilegiés**, dom1, dom2 etc., appelés communément les **domU**,

Restrictions

- Xen et **tous les domains** doivent utiliser la même architecture processeur. Vous ne pouvez pas lancer un dom0 en 64 bits et des domU en 32 bits,
- Xen est incompatible avec certains pilotes matériels binaires pour des cartes graphiques et cartes Wifi,
- Xen ne fournit pas de fonctions de son aux domU,
- Xen ne prend en charge l'ACPI et l'APM que d'une manière limitée,
- Xen ne peut pas utiliser les cartes PCMCIA et CardBus,
- Le dom0 ne peut fonctionner que sur un système Linux ou NetBSD,
- Xen est connu pour ses problèmes avec SELinux et AppArmor.

Gestion du matériel

- Xen gère le(s) **processeur(s)** et la **mémoire** pour le dom0 et les domU,
- Le dom0 gère pour lui-même le réseau et l'accès aux disques IDE/ATA/SATA,
- Les domU utilisent des pilotes Xen pour l'accès au réseau et aux disques IDE/ATA/SATA,
- Le dom0 gère les autres composants tels les cartes graphiques, cartes son, PCI, USB etc.,
- Les domU n'ont pas accès aux autres composants sauf dans un cas spécial.

Accès au réseau

- Le noyau du Dom0 charge les modules et initialise eth0 etc.,
 - A la fin du processus InitV, xen est démarré,
 - eth0 est renommé en **peth0**,
 - Le pont virtuel **xenbr0** est créé,
 - Le pont établit la connexion aux interfaces virtuelles, **vifu.i** où **u** représente le numéro du domU et **i** représente le numéro de l'interface,
 - Xen met à la disposition du dom0 les interfaces **vif0.0**, **vif0.1** etc.,
 - Le dom 0 voit l'interface **vif0.0** comme **eth0**,
 - Le dom1 voit l'interface **vif1.0** comme **eth0**,
 - etc..
-



Important - Le réseau de Xen est configuré dans le fichier `/etc/xen/xend-config.sxp`.

Accès au disque dur

- Le dom0 accède normalement au disque dur par son module habituel,
- Les domU accèdent au disque via le module Xen **xenblk**,

Sortie graphique

- Sauf cas spécifique, seul le dom0 a accès à la carte graphique,
- **SDL** (*Simple DirectMedia Layer*) peut être utilisé pour afficher le système graphique d'un domU dans une fenêtre de dom0,
- **VNC** peut aussi être utilisé pour afficher le système graphique d'un domU dans une fenêtre de dom0.



A faire - Consultez le site web de Xen - <http://www.xen.org/> pour vous familiariser avec les derniers développements de ce produit.

Installer le dom0

Installation de Xen

Installez le dépôt Xen :

```
[root@centos6 ~] yum install http://au1.mirror.crc.id.au/repo/kernel-xen-release-6.4.noarch.rpm
```

ensuite installez les paquets nécessaires :

```
[root@centos6 ~] yum install kernel-xen xen xen-libs libvirt python-virtinst libvirt-python virt-viewer virt-manager
```

Lors de l'installation des paquets, le fichier **/boot/grub/menu.lst** a été modifié :

[menu.lst](#)

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/mapper/vg_server-lv_root
#          initrd /initrd-[generic-]version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (3.11.9-1.el6xen.x86_64)
    root (hd0,0)
    kernel /xen.gz dom0_mem=1024M cpufreq=xen dom0_max_vcpus=1 dom0_vcpus_pin
    module /vmlinuz-3.11.9-1.el6xen.x86_64 ro root=/dev/mapper/vg_server-lv_root rd_LVM_LV=vg_server/lv_swap
rd_NO_LUKS KEYBOARDTYPE=pc KEYTABLE=fr LANG=en_US.UTF-8 rd_NO_MD rd_LVM_LV=vg_server/lv_root
SYSFONT=latarcyrheb-sun16 crashkernel=128M rd_NO_DM rhgb quiet
    module /initramfs-3.11.9-1.el6xen.x86_64.img
title CentOS (2.6.32-358.23.2.el6.x86_64)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-358.23.2.el6.x86_64 ro root=/dev/mapper/vg_server-lv_root
rd_LVM_LV=vg_server/lv_swap rd_NO_LUKS KEYBOARDTYPE=pc KEYTABLE=fr LANG=en_US.UTF-8 rd_NO_MD
rd_LVM_LV=vg_server/lv_root SYSFONT=latarcyrheb-sun16 crashkernel=128M rd_NO_DM rhgb quiet
    initrd /initramfs-2.6.32-358.23.2.el6.x86_64.img
title CentOS (2.6.32-358.el6.x86_64)
```

```
root (hd0,0)
kernel /vmlinuz-2.6.32-358.el6.x86_64 ro root=/dev/mapper/vg_server-lv_root rd_LVM_LV=vg_server/lv_swap
rd_NO_LUKS KEYBOARDTYPE=pc KEYTABLE=fr LANG=en_US.UTF-8 rd_NO_MD rd_LVM_LV=vg_server/lv_root
SYSFONT=latarcyrheb-sun16 crashkernel=128M rd_NO_DM rhgb quiet
initrd /initramfs-2.6.32-358.el6.x86_64.img
```

Redémarrez la machine. Lors du redémarrage de la machine, vous devez observer le démarrage de Xen :

[illegible]

```
(XEN) Xen version 4.2.3 (mockbuild@crc.id.au) (gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-3)) Fri Nov 22 14:21:46
EST 2013
```

(XEN) Latest ChangeSet: unavailable

(XEN) Bootloader: GNU GRUB 0.97

```
(XEN) Command line: dom0_mem=1024M cpufreq=xen dom0_max_vcpus=1 dom0_vcpus_pin
```

(XEN) Video information:

(XEN) VGA is text mode 80x25, font 8x16

```
(XEN) VBE/DDC methods: V2; EDID transfer time: 1 seconds
```

```
(XEN) Disc information:
```

```
(XEN) Found 1 MBR signatures
```

```
(XEN) Found 1 EDD information structures
```

(XEN) Xen-e820 RAM map:

```
(XEN) 000000000000000000 - 00000000000009d800 (usable)
```

(XEN) 0000000000009d800 - 000000000000a0000 (reserved)

(XEN) 000000000000e0000 - 00000000000100000 (reserved)

(XEN) 00000000000100000 - 00000000020000000 (usable)

(XEN) 0000000020000000 - 0000000020200000 (reserved)

(XEN) 0000000020200000 - 0000000040004000 (usable)

(XEN) 0000000040004000 - 0000000040005000 (reserved)

(XEN) 0000000040005000 - 00000000c90d0000 (usable)

```
(XEN) 00000000c90d0000 - 00000000c9113000 (ACPI NVS)
(XEN) 00000000c9113000 - 00000000c951d000 (usable)
(XEN) 00000000c951d000 - 00000000c97fb000 (reserved)
(XEN) 00000000c97fb000 - 00000000c9eef000 (usable)
(XEN) 00000000c9eef000 - 00000000c9ef4000 (reserved)
(XEN) 00000000c9ef4000 - 00000000c9f0b000 (usable)
(XEN) 00000000c9f0b000 - 00000000ca50c000 (ACPI NVS)
(XEN) 00000000ca50c000 - 00000000ca50f000 (reserved)
(XEN) 00000000ca50f000 - 00000000ca522000 (usable)
(XEN) 00000000ca522000 - 00000000ca528000 (reserved)
(XEN) 00000000ca528000 - 00000000ca52a000 (usable)
(XEN) 00000000ca52a000 - 00000000ca538000 (reserved)
(XEN) 00000000ca538000 - 00000000ca68b000 (usable)
(XEN) 00000000ca68b000 - 00000000ca68f000 (reserved)
(XEN) 00000000ca68f000 - 00000000ca6d8000 (usable)
(XEN) 00000000ca6d8000 - 00000000ca6f8000 (reserved)
(XEN) 00000000ca6f8000 - 00000000ca6fb000 (usable)
(XEN) 00000000ca6fb000 - 00000000ca6fd000 (reserved)
(XEN) 00000000ca6fd000 - 00000000ca714000 (usable)
(XEN) 00000000ca714000 - 00000000ca71a000 (reserved)
(XEN) 00000000ca71a000 - 00000000ca721000 (usable)
(XEN) 00000000ca721000 - 00000000ca722000 (reserved)
(XEN) 00000000ca722000 - 00000000ca73d000 (usable)
(XEN) 00000000ca73d000 - 00000000ca742000 (reserved)
(XEN) 00000000ca742000 - 00000000ca76e000 (usable)
(XEN) 00000000ca76e000 - 00000000ca76f000 (reserved)
(XEN) 00000000ca76f000 - 00000000ca77f000 (usable)
(XEN) 00000000ca77f000 - 00000000ca7a4000 (reserved)
(XEN) 00000000ca7a4000 - 00000000ca7b7000 (usable)
(XEN) 00000000ca7b7000 - 00000000ca7b8000 (reserved)
(XEN) 00000000ca7b8000 - 00000000ca7b9000 (usable)
(XEN) 00000000ca7b9000 - 00000000ca7bb000 (reserved)
(XEN) 00000000ca7bb000 - 00000000ca7bc000 (usable)
(XEN) 00000000ca7bc000 - 00000000ca7c1000 (reserved)
```

```
(XEN) 00000000ca7c1000 - 00000000ca7d7000 (usable)
(XEN) 00000000ca7d7000 - 00000000cad7a000 (reserved)
(XEN) 00000000cad7a000 - 00000000caffa000 (ACPI NVS)
(XEN) 00000000caffa000 - 00000000cafff000 (ACPI data)
(XEN) 00000000cafff000 - 00000000cb000000 (usable)
(XEN) 00000000cbc00000 - 00000000cfe00000 (reserved)
(XEN) 00000000fc000000 - 00000000fc000000 (reserved)
(XEN) 00000000fec00000 - 00000000fec01000 (reserved)
(XEN) 00000000fed00000 - 00000000fed04000 (reserved)
(XEN) 00000000fed1c000 - 00000000fed20000 (reserved)
(XEN) 00000000fee00000 - 00000000fee01000 (reserved)
(XEN) 00000000ff000000 - 0000000100000000 (reserved)
(XEN) 0000000100000000 - 00000001af200000 (usable)
(XEN) ACPI: RSDP 000F0490, 0024 (r2 _ASUS_)
(XEN) ACPI: XSDT CAFCD078, 0074 (r1 _ASUS_ Notebook 1072009 AMI 10013)
(XEN) ACPI: FACP CAFE0030, 00F4 (r4 _ASUS_ Notebook 1072009 AMI 10013)
(XEN) ACPI: DSDT CAFCD188, 12EA5 (r2 _ASUS_ Notebook 13 INTL 20091112)
(XEN) ACPI: FACS CAFF7F80, 0040
(XEN) ACPI: APIC CAFE0128, 0092 (r3 _ASUS_ Notebook 1072009 AMI 10013)
(XEN) ACPI: FPDT CAFE01C0, 0044 (r1 _ASUS_ Notebook 1072009 AMI 10013)
(XEN) ACPI: ECDT CAFE0208, 00C1 (r1 _ASUS_ Notebook 1072009 AMI. 5)
(XEN) ACPI: MCFG CAFE02D0, 003C (r1 _ASUS_ Notebook 1072009 MSFT 97)
(XEN) ACPI: SLIC CAFE0310, 0176 (r1 _ASUS_ Notebook 1072009 ASUS 1)
(XEN) ACPI: HPET CAFE0488, 0038 (r1 _ASUS_ Notebook 1072009 AMI. 5)
(XEN) ACPI: SSDT CAFE04C0, 090A (r1 PmRef Cpu0Ist 3000 INTL 20051117)
(XEN) ACPI: SSDT CAFE0DD0, 0A92 (r1 PmRef CpuPm 3000 INTL 20051117)
(XEN) ACPI: BGRT CAFE1868, 0038 (r0 _ASUS_ Notebook 1072009 ASUS 10013)
(XEN) System RAM: 6029MB (6174552kB)
(XEN) Domain heap initialised
(XEN) ACPI: 32/64X FACS address mismatch in FADT - caff7f80/0000000000000000, using 32
(XEN) Processor #0 7:10 APIC version 21
(XEN) Processor #2 7:10 APIC version 21
(XEN) Processor #4 7:10 APIC version 21
(XEN) Processor #6 7:10 APIC version 21
```

```
(XEN) Processor #1 7:10 APIC version 21
(XEN) Processor #3 7:10 APIC version 21
(XEN) Processor #5 7:10 APIC version 21
(XEN) Processor #7 7:10 APIC version 21
(XEN) IOAPIC[0]: apic_id 2, version 32, address 0xfec00000, GSI 0-23
(XEN) Enabling APIC mode: Flat. Using 1 I/O APICs
(XEN) Not enabling x2APIC: depends on iommu_supports_eim.
(XEN) Using scheduler: SMP Credit Scheduler (credit)
(XEN) Detected 2294.844 MHz processor.
(XEN) Initing memory sharing.
(XEN) xstate_init: using cntxt_size: 0x340 and states: 0x7
(XEN) I/O virtualisation disabled
(XEN) Enabled directed EOI with ioapic_ack_old on!
(XEN) ENABLING IO-APIC IRQs
(XEN) -> Using old ACK method
(XEN) Platform timer is 14.318MHz HPET
(XEN) Allocated console ring of 16 KiB.
(XEN) VMX: Supported advanced features:
(XEN) - APIC MMIO access virtualisation
(XEN) - APIC TPR shadow
(XEN) - Extended Page Tables (EPT)
(XEN) - Virtual-Processor Identifiers (VPID)
(XEN) - Virtual NMI
(XEN) - MSR direct-access bitmap
(XEN) - Unrestricted Guest
(XEN) HVM: ASIDs enabled.
(XEN) HVM: VMX enabled
(XEN) HVM: Hardware Assisted Paging (HAP) detected
(XEN) HVM: HAP page sizes: 4kB, 2MB
(XEN) Brought up 8 CPUs
(XEN) *** LOADING DOMAIN 0 ***
(XEN) Xen kernel: 64-bit, lsb, compat32
(XEN) Dom0 kernel: 64-bit, PAE, lsb, paddr 0x1000000 -> 0x1fc5000
(XEN) PHYSICAL MEMORY ARRANGEMENT:
```

```
(XEN) Dom0 alloc.: 00000001a0000000->00000001a4000000 (233661 pages to be allocated)
(XEN) Init. ramdisk: 00000001ac2bd000->00000001af1ff800
(XEN) VIRTUAL MEMORY ARRANGEMENT:
(XEN) Loaded kernel: ffffffff81000000->fffffff81fc5000
(XEN) Init. ramdisk: ffffffff81fc5000->fffffff84f07800
(XEN) Phys-Mach map: ffffffff84f08000->fffffff85108000
(XEN) Start info: ffffffff85108000->fffffff851084b4
(XEN) Page tables: ffffffff85109000->fffffff85136000
(XEN) Boot stack: ffffffff85136000->fffffff85137000
(XEN) TOTAL: ffffffff80000000->fffffff85400000
(XEN) ENTRY ADDRESS: ffffffff8184c1e0
(XEN) Dom0 has maximum 1 VCPUs
(XEN) Scrubbing Free RAM: .....done.
(XEN) Initial low memory virq threshold set at 0x4000 pages.
(XEN) Std. Loglevel: Errors and warnings
(XEN) Guest Loglevel: Nothing (Rate-limited: Errors and warnings)
(XEN) Xen is relinquishing VGA console.
(XEN) *** Serial input -> DOM0 (type 'CTRL-a' three times to switch input to Xen)
(XEN) Freed 272kB init memory.
(XEN) traps.c:2592:d0 Domain attempted WRMSR 000000000000001fc from 0x0000000000014005f to 0x0000000000014005d.
(XEN) traps.c:2592:d0 Domain attempted WRMSR 0000000000000079 from 0x0000000000000000 to 0xfffffc90008847030.
(XEN) traps.c:2592:d0 Domain attempted WRMSR 0000000000000079 from 0x0000000000000000 to 0xfffffc90008870030.
```



Important - Cette même information peut être visualisée à tout moment grâce à la commande **xm dmesg**.

Vérifier le Fonctionnement de Dom0

Vérifiez maintenant que votre dom0 est bien démarré :

```
[root@centos6 ~]# xm list
```

Name	ID	Mem	VCPUs	State	Time(s)
Domain-0	0	1024	1	r-----	231.8

La colonne **State** ou statut peut prendre 6 valeurs :

- **r** - running,
 - Le domain est fonctionnel,
- **b** - blocked,
 - Le domain est bloqué, n'est pas fonctionnel où ne peut pas être démarré. Cette situation peut arriver quand la machine virtuelle est en attente pour une E/S ou bien quand elle est au repos par manque de tâches à accomplir,
- **p** - paused
 - Le domain est en pause. Notez cependant que le domain consomme toujours les ressources système qui lui sont confiées. Par contre l'hyperviseur Xen ne gère pas le domain,
- **s** - shutdown,
 - La machine virtuelle a demandé d'être arrêtée, redémarrée ou suspendue et le domain est en cours de destruction,
- **c** - crashed,
 - Le domain a crashé,
- **d** - dying,
 - Le domain est mourant mais n'est pas encore arrêté complètement.

L'option **-long** nous donne plus d'information sur l'état et la configuration des domains :

```
[root@centos6 ~]# xm list --long
(domain
  (domid 0)
  (cpu_weight 256)
  (cpu_cap 0)
  (pool_name Pool-0)
  (bootloader '')
  (vcpus 1)
  (cpus ((0)))
  (on_poweroff destroy)
  (on_crash restart)
  (uuid 00000000-0000-0000-0000-000000000000))
```

```
(bootloader_args '')
(name Domain-0)
(on_reboot restart)
(maxmem 16777215)
(memory 1024)
(shadow_memory 0)
(features '')
(on_xend_start ignore)
(on_xend_stop ignore)
(cpu_time 237.215115864)
(online_vcpus 1)
(image (linux (kernel '')) (superpages 0) (nomigrate 0) (tsc_mode 0)))
(status 2)
(state r-----)
)
```

Xen - Configuration

Syntaxe du fichier de configuration du dom0

Le fichier principal de configuration de xen est **/etc/xen/xend-config.sxp** :

[xend-config.sxp](#)

```
# -*- sh -*-

#
# Xend configuration file.
#

# This example configuration is appropriate for an installation that
```

```
# utilizes a bridged network configuration. Access to xend via http
# is disabled.

# Commented out entries show the default for that entry, unless otherwise
# specified.

#(logfile /var/log/xen/xend.log)
#(loglevel DEBUG)

# Uncomment the line below. Set the value to flask, acm, or dummy to
# select a security module.

#(xsm_module_name dummy)

# The Xen-API server configuration.
#
# This value configures the ports, interfaces, and access controls for the
# Xen-API server. Each entry in the list starts with either unix, a port
# number, or an address:port pair. If this is "unix", then a UDP socket is
# opened, and this entry applies to that. If it is a port, then Xend will
# listen on all interfaces on that TCP port, and if it is an address:port
# pair, then Xend will listen on the specified port, using the interface with
# the specified address.
#
# The subsequent string configures the user-based access control for the
# listener in question. This can be one of "none" or "pam", indicating either
# that users should be allowed access unconditionally, or that the local
# Pluggable Authentication Modules configuration should be used. If this
# string is missing or empty, then "pam" is used.
#
# The final string gives the host-based access control for that listener. If
# this is missing or empty, then all connections are accepted. Otherwise,
# this should be a space-separated sequence of regular expressions; any host
# with a fully-qualified domain name or an IP address that matches one of
```

```
# these regular expressions will be accepted.
#
# Example: listen on TCP port 9363 on all interfaces, accepting connections
# only from machines in example.com or localhost, and allow access through
# the unix domain socket unconditionally:
#
# (xen-api-server ((9363 pam '^localhost$ example\\.com$')
#                  (unix none)))
#
# Optionally, the TCP Xen-API server can use SSL by specifying the private
# key and certificate location:
#
# (9367 pam '' xen-api.key xen-api.crt)
#
# Default:
# (xen-api-server ((unix)))

#(xend-http-server no)
#(xend-unix-server yes)
#(xend-tcp-xmlrpc-server no)
#(xend-unix-xmlrpc-server yes)
#(xend-relocation-server no)
#(xend-relocation-server yes)
#(xend-relocation-ssl-server no)
#(xend-udev-event-server no)

#(xend-unix-path /var/lib/xend/xend-socket)

# Address and port xend should use for the legacy TCP XMLRPC interface,
# if xend-tcp-xmlrpc-server is set.
#(xend-tcp-xmlrpc-server-address 'localhost')
#(xend-tcp-xmlrpc-server-port 8006)
```

```
# SSL key and certificate to use for the legacy TCP XMLRPC interface.
# Setting these will mean that this port serves only SSL connections as
# opposed to plaintext ones.
#(xend-tcp-xmlrpc-server-ssl-key-file  xmlrpc.key)
#(xend-tcp-xmlrpc-server-ssl-cert-file xmlrpc.crt)

# Port xend should use for the HTTP interface, if xend-http-server is set.
#(xend-port 8000)

# Port xend should use for the relocation interface, if xend-relocation-server
# is set.
#(xend-relocation-port 8002)

# Port xend should use for the ssl relocation interface, if
# xend-relocation-ssl-server is set.
#(xend-relocation-ssl-port 8003)

# SSL key and certificate to use for the ssl relocation interface, if
# xend-relocation-ssl-server is set.
#(xend-relocation-server-ssl-key-file  xmlrpc.key)
#(xend-relocation-server-ssl-cert-file xmlrpc.crt)

# Whether to use ssl as default when relocating.
#(xend-relocation-ssl no)

# Address xend should listen on for HTTP connections, if xend-http-server is
# set.
# Specifying 'localhost' prevents remote connections.
# Specifying the empty string '' (the default) allows all connections.
#(xend-address '')
#(xend-address localhost)
```

```
# Address xend should listen on for relocation-socket connections, if
# xend-relocation-server is set.
# Meaning and default as for xend-address above.
# Also, interface name is allowed (e.g. eth0) there to get the
# relocation address to be bound on.
#(xend-relocation-address '')

# The hosts allowed to talk to the relocation port. If this is empty (the
# default), then all connections are allowed (assuming that the connection
# arrives on a port and interface on which we are listening; see
# xend-relocation-port and xend-relocation-address above). Otherwise, this
# should be a space-separated sequence of regular expressions. Any host with
# a fully-qualified domain name or an IP address that matches one of these
# regular expressions will be accepted.
#
# For example:
# (xend-relocation-hosts-allow '^localhost$ ^.*\\.example\\.org$')
#
#(xend-relocation-hosts-allow '')
(xend-relocation-hosts-allow '^localhost$ ^localhost\\.localdomain$')

# The limit (in kilobytes) on the size of the console buffer
#(console-limit 1024)

##
# To bridge network traffic, like this:
#
# dom0: ----- bridge -> real eth0 -> the network
#           |
# domU: fake eth0 -> vifN.0 -+
#
# use
#
# (network-script network-bridge)
```

```
#
# Your default ethernet device is used as the outgoing interface, by default.
# To use a different one (e.g. eth1) use
#
# (network-script 'network-bridge netdev=eth1')
#
# The bridge is named eth0, by default (yes, really!)
#

# It is normally much better to create the bridge yourself in
# /etc/network/interfaces. network-bridge start does nothing if you
# already have a bridge, and network-bridge stop does nothing if the
# default bridge name (normally eth0) is not a bridge. See
# bridge-utils-interfaces(5) for full information on the syntax in
# /etc/network/interfaces, but you probably want something like this:
#     iface xenbr0 inet static
#         address [etc]
#         netmask [etc]
#         [etc]
#         bridge_ports eth0
#
# To have network-bridge create a differently-named bridge, use:
# (network-script 'network-bridge bridge=<name>')
#
# It is possible to use the network-bridge script in more complicated
# scenarios, such as having two outgoing interfaces, with two bridges, and
# two fake interfaces per guest domain. To do things like this, write
# yourself a wrapper script, and call network-bridge from it, as appropriate.
#
#(network-script network-bridge)
(network-script /bin/true)

# The script used to control virtual interfaces. This can be overridden on a
# per-vif basis when creating a domain or a configuring a new vif. The
```

```
# vif-bridge script is designed for use with the network-bridge script, or
# similar configurations.
#
# If you have overridden the bridge name using
# (network-script 'network-bridge bridge=<name>') then you may wish to do the
# same here. The bridge name can also be set when creating a domain or
# configuring a new vif, but a value specified here would act as a default.
#
# If you are using only one bridge, the vif-bridge script will discover that,
# so there is no need to specify it explicitly. The default is to use
# the bridge which is listed first in the output from brctl.
#
(vif-script vif-bridge)

## Use the following if network traffic is routed, as an alternative to the
# settings for bridged networking given above.
#(network-script network-route)
#(vif-script      vif-route)

## Use the following if network traffic is routed with NAT, as an alternative
# to the settings for bridged networking given above.
#(network-script network-nat)
#(vif-script      vif-nat)

# dom0-min-mem is the lowest permissible memory level (in MB) for dom0.
# This is a minimum both for auto-ballooning (as enabled by
# enable-dom0-ballooning below) and for xm mem-set when applied to dom0.
(dom0-min-mem 256)

# Whether to enable auto-ballooning of dom0 to allow domUs to be created.
# If enable-dom0-ballooning = no, dom0 will never balloon out.
(enable-dom0-ballooning yes)
```

```
# 32-bit paravirtual domains can only consume physical
# memory below 168GB. On systems with memory beyond that address,
# they'll be confined to memory below 128GB.
# Using total_available_memory (in GB) to specify the amount of memory reserved
# in the memory pool exclusively for 32-bit paravirtual domains.
# Additionally you should use dom0_mem = <-Value> as a parameter in
# xen kernel to reserve the memory for 32-bit paravirtual domains, default
# is "0" (0GB).
(total_available_memory 0)

# In SMP system, dom0 will use dom0-cpus # of CPUs
# If dom0-cpus = 0, dom0 will take all cpus available
(dom0-cpus 0)

# Whether to enable core-dumps when domains crash.
#(enable-dump no)

# The tool used for initiating virtual TPM migration
#(external-migration-tool '')

# The interface for VNC servers to listen on. Defaults
# to 127.0.0.1 To restore old 'listen everywhere' behaviour
# set this to 0.0.0.0
#(vnc-listen '127.0.0.1')

# The default password for VNC console on HVM domain.
# Empty string is no authentication.
(vncpasswd '')

# The VNC server can be told to negotiate a TLS session
# to encryption all traffic, and provide x509 cert to
# clients enabling them to verify server identity. The
# GTK-VNC widget, virt-viewer, virt-manager and VeNCrypt
```

```
# all support the VNC extension for TLS used in QEMU. The
# TightVNC/RealVNC/UltraVNC clients do not.
#
# To enable this create x509 certificates / keys in the
# directory ${XEN_CONFIG_DIR} + vnc
#
#  ca-cert.pem          - The CA certificate
#  server-cert.pem      - The Server certificate signed by the CA
#  server-key.pem       - The server private key
#
# and then uncomment this next line
# (vnc-tls 1)

# The certificate dir can be pointed elsewhere..
#
# (vnc-x509-cert-dir vnc)

# The server can be told to request & validate an x509
# certificate from the client. Only clients with a cert
# signed by the trusted CA will be able to connect. This
# is more secure the password auth alone. Passwd auth can
# used at the same time if desired. To enable client cert
# checking uncomment this:
#
# (vnc-x509-verify 1)

# The default keymap to use for the VM's virtual keyboard
# when not specififed in VM's configuration
#(keymap 'en-us')

# Script to run when the label of a resource has changed.
#(resource-label-change-script '')

# Rotation count of qemu-dm log file.
```

```
$(qemu-dm-logrotate-count 10)

# Path where persistent domain configuration is stored.
# Default is /var/lib/xend/domains/
$(xend-domains-path /var/lib/xend/domains)

# Number of seconds xend will wait for device creation and
# destruction
$(device-create-timeout 100)
$(device-destroy-timeout 100)

# When assigning device to HVM guest, we use the strict check for HVM guest by
# default. (For PV guest, we use loose check automatically if necessary.)
# When we assign device to HVM guest, if we meet with the co-assignment
# issues or the ACS issue, we could try changing the option to 'no' -- however,
# we have to realize this may incur security issue and we can't make sure the
# device assignment could really work properly even after we do this.
$(pci-passthrough-strict-check yes)

# If we have a very big scsi device configuration, start of xend is slow,
# because xend scans all the device paths to build its internal PSCSI device
# list. If we need only a few devices for assigning to a guest, we can reduce
# the scan to this device. Set list list of device paths in same syntax like in
# command lsscsi, e.g. ('16:0:0:0' '15:0')
# (pscsi-device-mask ('*'))
```

A l'installation les directives suivantes sont actives :

```
(xend-unix-server yes)
(xend-unix-path /var/lib/xend/xend-socket)
(xend-relocation-hosts-allow '^localhost$ ^localhost\\.localdomain$')
(network-script network-bridge)
(vif-script vif-bridge)
```

```
(dom0-min-mem 256)
(dom0-cpus 0)
(vncpasswd '')
(keymap 'en-us')
```

Syntaxe du fichier de configuration du domU

Chaque domU est configuré par un fichier, par exemple le fichier **/etc/xen/xen100.fenestros.loc** :

```
name = "xen100.fenestros.loc"
uuid = "635b1d73-a9ed-6178-464c-8f05ab464e76"
maxmem = 512
memory = 512
vcpus = 1
bootloader = "/usr/bin/pygrub"
on_poweroff = "destroy"
on_reboot = "restart"
on_crash = "restart"
vfb = [ "type=vnc,vncunused=1,keymap=fr" ]
disk = [ "phy:/dev/vg0/lv0,xvda,w" ]
vif = [ "mac=00:16:36:00:13:90,bridge=xenbr0,script=vif-bridge" ]
```



Important - Ce fichier est au format d'un script **Python** et son interprétation se fait comme un script python standard. Il existe un autre format de fichier de configuration dit **SXP** ou **S-expressions**. Nous détaillerons ce type de format lors de l'étude de virsh.

Les directives importantes de ce fichier sont :

name

Cette directive indique le nom du domain. Elle prend la forme suivante :

```
name = 'nom'
```

Exemple :

```
name = "xen100.fenestros.loc"
```

UUID

L'UUID est un *Universal Unique Identifier* pour la machine virtuelle. Elle prend la forme suivante :

```
uuid="<128-bit Hexdecimal>"
```

Exemple :

```
uuid = "635b1d73-a9ed-6178-464c-8f05ab464e76"
```

maxmem

Cette directive permet de définir la quantité de mémoire vive maximum que pourra avoir le domU lors d'une augmentation en cours de fonctionnement. Elle prend la forme suivante :

```
maxmem = 'n'
```

Exemple :

```
maxmem = 512
```

memory

Cette directive stipule la quantité de mémoire en Mo réservée au domain. Elle prend la forme suivante :

```
memory = 'n'
```

Exemple :

```
memory      = '512'
```

vcpus

Cette directive indique le nombre de processeurs virtuels associés au domain. Par défaut sa valeur est de **1**. Elle prend la forme suivante :

```
vcpus = 'n'
```

Exemple :

```
vcpus = 1
```

bootloader

Cette directive spécifie le chemin du gestionnaire d'amorçage pour le DomU. Elle prend la forme suivante :

```
bootloader= "<chemin_du_gestionnaire/pygrub>=
```

Exemple :

```
bootloader = "/usr/bin/pygrub"
```

Important - Notez que dans certaines configurations, pygrub n'est pas utilisé en tant que gestionnaire d'amorçage au quel cas nous trouvons des directives telles que les suivantes dans le fichier de configuration :



```
kernel = "/boot/vmlinuz-2.6.18-164.el5xen"  
ramdisk = "/boot/initrd.img-2.6.18-164.el5xen"
```

En effet la différence entre les deux systèmes est simple. Dans le cas de l'utilisation de pygrub, c'est le noyau et l'initrd du *DomU* qui sont utilisés pour démarrer celui-ci. Dans le cas des directives **kernel** et **ramdisk** celles-ci indiquent l'utilisation du noyau et de l'initrd du *Dom0* pour démarrer le *DomU*.

on_poweroff

Cette directive indique le comportement de domain lors d'un poweroff. Elle prend la forme suivante :

```
on_poweroff="comportement"
```

Le comportement peut être :

- destroy
- restart
- rename-restart
- preserve

Exemple :

```
on_poweroff = "destroy"
```

on_reboot

Cette directive indique le comportement de domain lors d'un reboot. Elle prend la forme suivante :

```
on_reboot="comportement"
```

Le comportement peut être :

- destroy
- restart
- rename-restart
- preserve

Exemple :

```
on_reboot = "restart"
```

on_crash

Cette directive indique le comportement de domain lors d'un crash. Elle prend la forme suivante :

```
on_crash="comportement"
```

Le comportement peut être :

- destroy
- restart
- rename-restart
- preserve

Exemple :

```
on_crash = "restart"
```

vfb

Cette directive définit le type de **Virtual Frame Buffer** utilisé par le programme **virt-manager**. Elle prend la forme suivante :

```
vfb = [ "type=type,paramètre1,paramètre2,..." ]
```

Exemple :

```
vfb = [ "type=vnc,vncunused=1,keymap=fr" ]
```

Le paramètre **vncunused=N** démarre le serveur vnc sur le port **5900+N**. Dans le cas donc de l'exemple c'est le port 5901 qui est à l'écoute des connexions entrantes.

Le paramètre **keymap=code** définit le **keymap** pour la machine virtuelle. Les keymaps sont stockés dans le répertoire **/usr/share/xen/qemu/keymaps/**

```
[root@centos6 ~]# ls -l /usr/share/xen/qemu/keymaps/
total 280
-rw-r--r-- 1 root root 2239 mai 18 2007 ar
-rw-r--r-- 1 root root 2078 mai 18 2007 common
-rw-r--r-- 1 root root 2576 mai 18 2007 da
-rw-r--r-- 1 root root 2467 mai 18 2007 de
-rw-r--r-- 1 root root 2542 mai 18 2007 de-ch
-rw-r--r-- 1 root root 2581 mai 18 2007 en-gb
-rw-r--r-- 1 root root 609 mai 18 2007 en-us
-rw-r--r-- 1 root root 2236 mai 18 2007 es
-rw-r--r-- 1 root root 1121 mai 18 2007 et
-rw-r--r-- 1 root root 2685 mai 18 2007 fi
-rw-r--r-- 1 root root 907 mai 18 2007 fo
-rw-r--r-- 1 root root 2588 mai 18 2007 fr
```

```
-rw-r--r-- 1 root root 2876 mai 18 2007 fr-be
-rw-r--r-- 1 root root 934 mai 18 2007 fr-ca
-rw-r--r-- 1 root root 2471 mai 18 2007 fr-ch
-rw-r--r-- 1 root root 2699 mai 18 2007 hr
-rw-r--r-- 1 root root 1866 mai 18 2007 hu
-rw-r--r-- 1 root root 3075 mai 18 2007 is
-rw-r--r-- 1 root root 2462 mai 18 2007 it
-rw-r--r-- 1 root root 2131 sep 4 06:45 ja
-rw-r--r-- 1 root root 1096 mai 18 2007 lt
-rw-r--r-- 1 root root 2840 mai 18 2007 lv
-rw-r--r-- 1 root root 2366 mai 18 2007 mk
-rw-r--r-- 1 root root 296 sep 4 06:45 modifiers
-rw-r--r-- 1 root root 1107 mai 18 2007 nl
-rw-r--r-- 1 root root 43 mai 18 2007 nl-be
-rw-r--r-- 1 root root 2576 mai 18 2007 no
-rw-r--r-- 1 root root 2672 mai 18 2007 pl
-rw-r--r-- 1 root root 2450 mai 18 2007 pt
-rw-r--r-- 1 root root 1354 mai 18 2007 pt-br
-rw-r--r-- 1 root root 2554 mai 18 2007 ru
-rw-r--r-- 1 root root 2293 mai 18 2007 sl
-rw-r--r-- 1 root root 1013 mai 18 2007 sv
-rw-r--r-- 1 root root 3123 mai 18 2007 th
-rw-r--r-- 1 root root 2649 mai 18 2007 tr
```

disk

Cette directive décrit un ou plusieurs supports de données. Elle prend la forme suivante :

```
disk = ['chaine1', 'chaine2'...]
```

Chaque *chaine* prend la syntaxe suivante : type:périphérique_réel,périphérique_virtuel,mode.

Le **type** fait référence à un type de *device* dans le Dom0 et peut prendre une des valeurs suivantes :

- tap:aio,
- phy,
- file,

tandis que le **mode** fait référence au type d'accès au disque et peut prendre une des valeurs suivantes :

- r,
- w.

Exemple :

```
disk = [ "phy:/dev/vg0/lv0,xvda,w" ]
```

vif

Cette directive indique les interfaces réseaux. Elle prend la forme suivante :

```
vif = ['chaine1', 'chaine2' ...]
```

Si *chaine* est vide, le domain utilise du DHCP. Chaque *chaine* prend la syntaxe suivante : "type, mac, bridge, ip"

Exemple :

```
vif = [ "mac=00:16:36:00:13:90,bridge=xenbr0,script=vif-bridge" ]
```

Il est à noter que pour les HVM, la valeur de type doit être **ioemu** :

Xen - Installer les domU

Avant de commencer, créez un répertoire pour stocker les machines virtuelles en format de fichiers :

```
[root@centos6 ~]# mkdir -p /home/xen/domains
```

L'outil de base pour la création d'invités sous CentOS est **virt-install**.

Un domU CentOS dans un fichier

Créez d'abord le répertoire pour notre machine virtuelle :

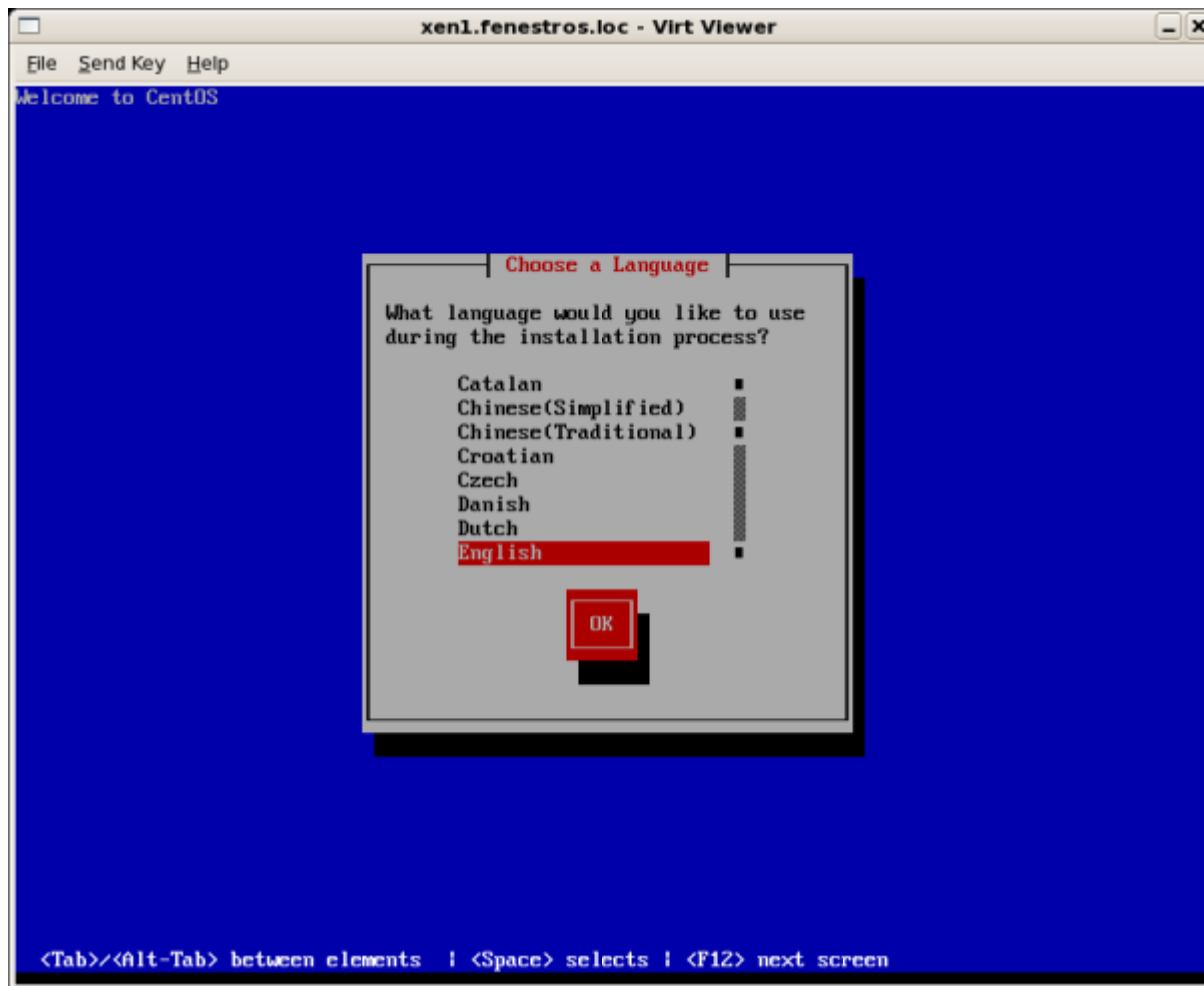
```
[root@centos6 ~]# mkdir /home/xen/domains/xen1.fenestros.loc/
```

Lancez ensuite virt-install d'une manière interactive avec l'option - **-prompt** et répondez aux questions posées :

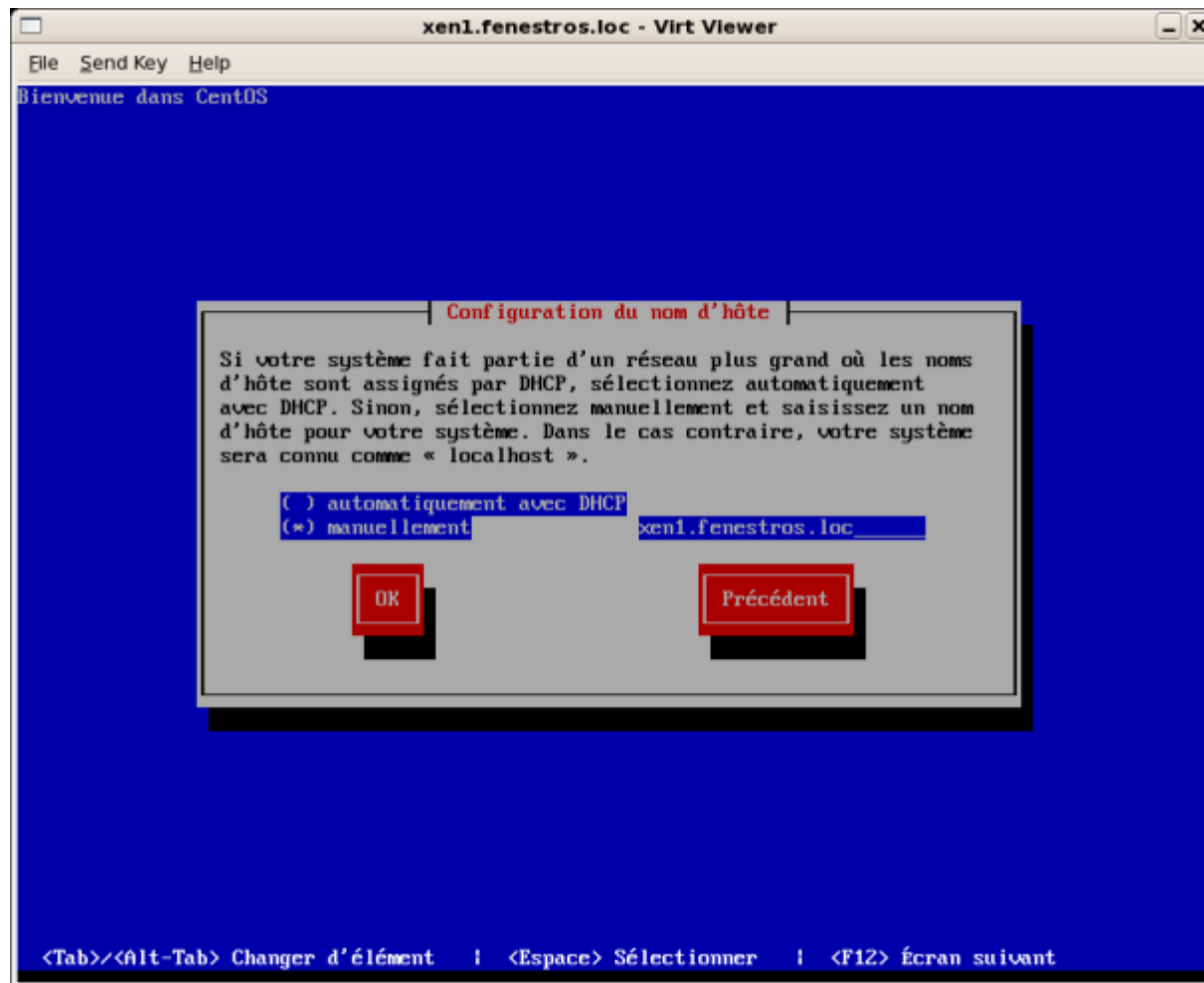
```
[root@centos6 ~]# virt-install --prompt
What is the name of your virtual machine? xen1.fenestros.loc
How much RAM should be allocated (in megabytes)? 256
What would you like to use as the disk (file path)? /home/xen/domains/xen1.fenestros.loc/disk.img
How large would you like the disk (/home/xen/domains/xen1.fenestros.loc/disk.img) to be (in gigabytes)? 6
What is the install URL? http://mirrors.prometeus.net/centos/5.10/isos/x86_64/
```

```
Démarrage de l'installation...
Récupération du fichier vmlinuz...
| 1.9 MB      00:04
Récupération du fichier initrd.img...
| 6.3 MB      00:12
Création du fichier de stockage...
| 4.0 GB      00:00
Création du domaine...
| 0 B         00:00
```

A ce stade, virt-install fait appel à **virt-viewer** pour l'installation. Vous obtiendrez donc une fenêtre similaire à celle-ci :



Procédez à l'installation de la machine virtuelle en DHCP et en spécifiant un nom de machine **xen1.fenestros.loc** :



A l'issue de l'installation, consultez le fichier `/etc/xen/xen1.fenestros.loc`. Vous devez obtenir un résultat similaire à celui-ci :

```
name = "xen1.fenestros.loc"
uuid = "33311a78-4a72-9242-5b63-7ab774b2aebc"
maxmem = 256
memory = 256
vcpus = 1
bootloader = "/usr/bin/pygrub"
```

```
on_poweroff = "destroy"
on_reboot = "restart"
on_crash = "restart"
vfb = [ "type=vnc,vncunused=1,keymap=fr" ]
disk = [ "tap:aio:/home/xen/domains/xen1.fenestros.loc/disk.img,xvda,w" ]
vif = [ "mac=00:16:36:09:a0:3e,bridge=xenbr0,script=vif-bridge" ]
```

Eteignez la machine virtuelle :

```
[root@centos6 ~]# xm destroy xen1.fenestros.loc
```

Enlevez la machine virtuelle de Xen :

```
[root@centos6 ~]# xm delete xen1.fenestros.loc
```



A faire - Vérifiez maintenant si l'image de xen1.fenestros.loc a été supprimée.

Créez maintenant le répertoire pour notre deuxième machine virtuelle :

```
[root@centos6 ~]# mkdir /home/xen/domains/xen10.fenestros.loc/
```

Lancez ensuite la commande **virt-install** avec des options en ligne de commande :

```
[root@centos6 ~]# virt-install --name=xen10.fenestros.loc --ram=512 --os-type=linux --os-variant=rhel5 --paravirt
--location=http://mirror.ovh.net/ftp.centos.org/5.4/os/i386/ --disk
path=/home/xen/domains/xen10.fenestros.loc/disk.img,size=6 --vnc --network network:default
```

Démarrage de l'installation...

Récupération du fichier .treeinfo...

Récupération du fichier vmlinuz...

| 413 B 00:00

| 2.1 MB 00:04

Récupération du fichier initrd.img...	6.6 MB	00:12
Création du fichier de stockage...	4.0 GB	00:00
Création du domaine...	0 B	00:00



A faire - Consultez le manuel de virt-install pour vous familiariser avec les options en ligne de commande.

Notez que virt-install lance de nouveau **virt-viewer** mais cette fois-ci en mode graphique. Poursuivez donc votre installation.

A l'issu de l'installation, consultez le fichier **/etc/xen/xen10.fenestros.loc**. Vous devez obtenir un résultat similaire à celui-ci :

```
name = "xen10.fenestros.loc"
uuid = "497a10b7-9b8c-c3d0-2549-2de61ff116f3"
maxmem = 512
memory = 512
vcpus = 1
bootloader = "/usr/bin/pygrub"
on_poweroff = "destroy"
on_reboot = "restart"
on_crash = "restart"
vfb = [ "type=vnc,vncunused=1,keymap=fr" ]
disk = [ "tap:aio:/home/xen/domains/xen10.fenestros.loc/disk.img,xvda,w" ]
vif = [ "mac=00:16:36:3c:3c:2a,bridge=virbr0,script=vif-bridge" ]
```

Utilisez la commande **xm** pour constater la présence de votre machine virtuelle :

```
[root@centos6 ~]# xm list
```

Name	ID	Mem(MiB)	VCPUs	State	Time(s)
Domain-0	0	2482	2	r-----	1174.0
xen10.fenestros.loc	3	511	1	-b----	65.0

Arrêtez maintenant la machine virtuelle et supprimez son répertoire :

```
[root@centos6 ~]# xm destroy 3  
[root@centos6 ~]# rm -rf /home/xen/domains/xen10.fenestros.loc/
```

Notez que cette fois-ci nous faisons référence à la machine virtuelle par son **#ID**.

HERE

Un domU CentOS dans un VG

Créer sur le disque dur de votre machine une partition d'au moins 8Go estampillée LVM (8e).

Dans le cas de l'exemple qui suit la partition concernée est **/dev/sda8**.

Pour initialiser le LVM saisissez la commande suivante :

```
# vgscan [Entrée]
```

Par exemple :

```
[root@centos6 ~]# vgscan  
Reading all physical volumes. This may take a while...
```

Pour créer le **Physical Volume** le paquetage **lvm2** contient la commande **pvccreate** :

```
# pvcreate /dev/sda8 [Entrée]
```

Vous obtiendrez un résultat similaire à celui-ci :

```
[root@centos6 ~]# pvcreate /dev/sda8  
Physical volume "/dev/sda8" successfully created
```

Pour contrôler les PV le paquetage **lvm2** contient la commande **pvdisplay** :

```
# pvdisplay /dev/sda8 [Entrée]
```

Vous obtiendrez un résultat similaire à celui-ci :

```
[root@centos6 ~]# pvdisplay /dev/sda8
"/dev/sda8" is a new physical volume of "8,36 GB"
--- NEW Physical volume ---
PV Name                /dev/sda8
VG Name
PV Size                8,36 GB
Allocatable            NO
PE Size (KByte)        0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                9sV8Aj-8Bcr-ycq6-4C95-cFLB-lgiY-e75egY
```

Pour créer un groupe de volumes dénommé **vg0**, le paquetage **lvm2** contient la commande **vgcreate** :

```
# vgcreate -s 4M vg0 /dev/sda8 [Entrée]
```

Vous obtiendrez un résultat similaire à celui-ci :

```
[root@centos6 ~]# vgcreate -s 4M vg0 /dev/sda8
Volume group "vg0" successfully created
```

Pour afficher les informations concernant **vg0**, le paquetage **lvm2** contient la commande **vgdisplay** :

```
# vgdisplay vg0 [Entrée]
```

Vous obtiendrez un résultat similaire à celui-ci :

```
[root@centos6 ~]# vgdisplay vg0
```

```
--- Volume group ---
VG Name                vg0
System ID
Format                 lvm2
Metadata Areas         1
Metadata Sequence No   1
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 0
Open LV                 0
Max PV                  0
Cur PV                 1
Act PV                  1
VG Size                 8,36 GB
PE Size                 4,00 MB
Total PE                2139
Alloc PE / Size         0 / 0
Free PE / Size          2139 / 8,36 GB
VG UUID                 nGDlB4-TMz8-xsk1-etfw-pmI1-Co4g-3w7CDn
```

Créez ensuite le volume logique **lv0** d'une taille de 7 000 Mo :

```
[root@centos6 ~]# lvcreate -L 7000 -n lv0 vg0
Logical volume "lv0" created
```

Utilisez ensuite la commande **lvdisplay** pour afficher les caractéristiques du volume logique :

```
[root@centos6 ~]# lvdisplay
--- Logical volume ---
LV Name                /dev/vg0/lv0
VG Name                vg0
LV UUID                WglPm4-mT3y-TmYi-pXl0-a2SQ-DkRv-e6VEaM
LV Write Access         read/write
```

```
LV Status          available
# open             0
LV Size            6,84 GB
Current LE         1750
Segments           1
Allocation          inherit
Read ahead sectors auto
- currently set to 256
Block device        253:0
```

Créez maintenant votre machine virtuelle avec la commande **virt-install** :

```
[root@centos6 ~]# virt-install --prompt
What is the name of your virtual machine? xen100.fenestros.loc
How much RAM should be allocated (in megabytes)? 512
What would you like to use as the disk (file path)? /dev/vg0/lv0
What is the install URL? http://mirror.ovh.net/ftp.centos.org/5.4/os/i386/

Démarrage de l'installation...
Récupération du fichier .treeinfo... | 413 B    00:00
Récupération du fichier vmlinuz...  | 2.1 MB   00:04
Récupération du fichier initrd.img... | 6.6 MB   00:12
Création du domaine...              | 0 B      00:01
```

Notez que la taille n'est plus demandée car celle-ci est définie par la taille du volume logique.

Xen - Administration

Outils en ligne de commande

La commande xm

Cette commande est la commande de base pour l'administration de Xen.

Usage: `xm <subcommand> [args]`

Control, list, and manipulate Xen guest instances.

xm full list of subcommands:

<code>console</code>	Attach to <Domain>'s console.
<code>create</code>	Create a domain based on <ConfigFile>.
<code>destroy</code>	Terminate a domain immediately.
<code>domid</code>	Convert a domain name to domain id.
<code>domname</code>	Convert a domain id to domain name.
<code>dump-core</code>	Dump core for a specific domain.
<code>list</code>	List information about all/some domains.
<code>mem-max</code>	Set the maximum amount reservation for a domain.
<code>mem-set</code>	Set the current memory usage for a domain.
<code>migrate</code>	Migrate a domain to another machine.
<code>pause</code>	Pause execution of a domain.
<code>reboot</code>	Reboot a domain.
<code>rename</code>	Rename a domain.
<code>restore</code>	Restore a domain from a saved state.
<code>save</code>	Save a domain state to restore later.
<code>shutdown</code>	Shutdown a domain.
<code>sysrq</code>	Send a sysrq to a domain.
<code>trigger</code>	Send a trigger to a domain.
<code>top</code>	Monitor a host and the domains in real time.
<code>unpause</code>	Unpause a paused domain.
<code>uptime</code>	Print uptime for a domain.
<code>vcpu-list</code>	List the VCPUs for a domain or all domains.
<code>vcpu-pin</code>	Set which CPUs a VCPU can use.

vcpu-set	Set the number of active VCPUs for allowed for the domain.
dmesg	Read and/or clear Xend's message buffer.
info	Get information about Xen host.
log	Print Xend log
serve	Proxy Xend XMLRPC over stdio.
sched-credit	Get/set credit scheduler parameters.
sched-sedf	Get/set EDF parameters.
block-attach	Create a new virtual block device.
block-detach	Destroy a domain's virtual block device.
block-list	List virtual block devices for a domain.
block-configure	Change block device configuration
network-attach	Create a new virtual network device.
network-detach	Destroy a domain's virtual network device.
network-list	List virtual network interfaces for a domain.
vtpm-list	List virtual TPM devices.
pci-attach	Insert a new pass-through pci device.
pci-detach	Remove a domain's pass-through pci device.
pci-list	List pass-through pci devices for a domain.
pci-list-assignable-devices	List all the assignable pci devices
vnet-list	List Vnets.
vnet-create	Create a vnet from ConfigFile.
vnet-delete	Delete a Vnet.
labels	List <type> labels for (active) policy.
addlabel	Add security label to domain.
rmlabel	Remove a security label from domain.
getlabel	Show security label for domain or resource.
dry-run	Test if a domain can access its resources.
resources	Show info for each labeled resource.
makepolicy	Build policy and create .bin/.map files.
loadpolicy	Load binary policy into hypervisor.
cfgbootpolicy	Add policy to boot configuration.
dumppolicy	Print hypervisor ACM state information.

```
<Domain> can either be the Domain Name or Id.  
For more help on 'xm' see the xm(1) man page.  
For more help on 'xm create' see the xmdomain.cfg(5) man page.
```

Les principales sous-commandes importantes sont :

block-attach

Format : `xm block-attach idréel virtuel mode`

Cette commande associe au domain le support de données *idréel* sous le nom *virtuel* en mode *ro* ou *rw*

block-detach

Format : `xm block-detach iddomain idpériphérique`

Cette commande supprime un support de données du domain.

block-list

Format : `xm block-list id`

Cette commande affiche la liste des périphériques attachés à un domain avec leurs *idpériphériques*.

console

Format : `xm console id`

Cette commande lance la console du domain passé en argument.

create

Format : `xm create [-c] fichier`

Cette commande démarre le domain indiqué par le fichier de configuration. L'option **c** démarre automatiquement la console.

destroy

Format : `xm destroy id`

Cette commande éteint la machine immédiatement.

dmesg

Format : `xm dmesg`

Cette commande affiche les messages de Xen lors de son démarrage.

info

Format : `xm info`

Cette commande affiche les informations sur Xen.

list

Format : `xm list`

Cette commande affiche tous les domaines lancés avec leur état :

- **r** - running - en cours d'exécution,
- **b** - blocked - bloqué ou en attente,
- **p** - paused - en pause,
- **s** - shutdown - en cours d'arrêt,
- **c** - crashed - planté,
- **d** - dying - mourant.

network-attach

Format : `xm network-attach id paramètre`

Cette commande associe une nouvelle interface réseau au domain.

network-detach

Format : `xm network-attach id domain idinterface`

Cette commande supprime l'interface.

network-list

Format : `xm network-list id`

Cette commande montre les interfaces du domain.

pause

Format : `xm pause id`

Cette commande met un domain en pause. En pause le domain consomme toujours de la mémoire mais pas de temps CPU.

shutdown

Format : `xm shutdown [-w] id`

Cette commande lance l'arrêt du domain. L'option **-w** indique que xm attend la fin de l'arrêt.

unpause

Format : `xm unpause id`

Cette commande relance un domain *id* après une pause.

La bibliothèque libvirt

La bibliothèque **libvirt** est une couche de gestion pour Xen. Afin de ce faciliter le travail dans un environnement hétérogène, elle peut aussi être utilisée pour gérer d'autre hyperviseurs tels :

- KVM,
- Qemu,
- OpenVZ,
- LXC,
- User Mode Linux,
- VirtualBox,
- VMware.

Fichiers de Configuration

Libvirt utilise des fichiers de configuration au format XML. Un exemple d'un tel fichier est :

```
<domain type='xen' id='2'>
  <name>client01</name>
  <uuid>d80c1821-4e28-bc5b-f22a-17c0a7355aa1</uuid>
  <memory>262144</memory>
  <currentMemory>262144</currentMemory>
  <vcpu>1</vcpu>
  <bootloader></bootloader>
  <os>
    <type>linux</type>
    <kernel>/live/image/guests/vmlinuz-2.6.27-11-server</kernel>
    <initrd>/live/image/guests/initrd.img-2.6.27-11-server</initrd>
    <cmdline>root=/dev/xvda1 aufs=tmpfs ro console=tty0 guestname=client01</cmdline>
  </os>
  <clock offset='utc' />
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/lib/xen-3.2-1/bin/qemu-dm</emulator>
    <disk type='file' device='disk'>
      <driver name='file' />
      <source file='/live/image/guests/ubuntu-8.10-desktop-i386.squashfs' />
      <target dev='xvda1' bus='xen' />
      <readonly />
    </disk>
    <interface type='bridge'>
      <mac address='16:a2:77:1a:b4:af' />
      <source bridge='eth0' />
      <target dev='vif2.0' />
    </interface>
    <console type='pty' tty='/dev/pts/2'>
      <source path='/dev/pts/2' />
    </console>
  </devices>
</domain>
```

```
<target port='0' />
</console>
<input type='mouse' bus='xen' />
<graphics type='vnc' port='5903' autoport='no' />
</devices>
</domain>
```

Pour créer un tel fichier il convient d'abord de démarrer la machine virtuelle avec la commande **xm** :

```
# xm create /etc/xen/xen100.fenestros.loc [Entrée]
```

```
[root@centos6 ~]# xm create /etc/xen/xen100.fenestros.loc
Using config file "/etc/xen/xen100.fenestros.loc".
Started domain xen100.fenestros.loc
```

Ensuite il convient d'utiliser la commande **virsh** de la bibliothèque afin de créer le fichier XML :

```
[root@centos6 ~]# virsh dumpxml xen100.fenestros.loc > /etc/xen/xen100.fenestros.loc.xml
```

Le fichier ainsi généré devrait être :

```
<domain type='xen' id='1'>
  <name>xen100.fenestros.loc</name>
  <uuid>635b1d73-a9ed-6178-464c-8f05ab464e76</uuid>
  <memory>524288</memory>
  <currentMemory>524288</currentMemory>
  <vcpu>1</vcpu>
  <bootloader>/usr/bin/pygrub</bootloader>
  <os>
    <type>linux</type>
    <kernel>/var/lib/xen/boot_kernel._u63kx</kernel>
    <initrd>/var/lib/xen/boot_ramdisk.TRdZV0</initrd>
    <cmdline>ro root=/dev/VolGroup00/LogVol00 rhgb quiet</cmdline>
  </os>
```

```
<clock offset='utc' />
<on_poweroff>destroy</on_poweroff>
<on_reboot>restart</on_reboot>
<on_crash>restart</on_crash>
<devices>
  <disk type='block' device='disk'>
    <driver name='phy' />
    <source dev='/dev/vg0/lv0' />
    <target dev='xvda' bus='xen' />
  </disk>
  <interface type='bridge'>
    <mac address='00:16:36:00:13:90' />
    <source bridge='xenbr0' />
    <script path='vif-bridge' />
    <target dev='vif1.0' />
  </interface>
  <console type='pty' tty='/dev/pts/2'>
    <source path='/dev/pts/2' />
    <target port='0' />
  </console>
  <input type='mouse' bus='xen' />
  <graphics type='vnc' port='5900' autoport='yes' keymap='fr' />
</devices>
</domain>
```

Il est ensuite nécessaire de déclarer la machine virtuelle auprès de virsh :

```
[root@centos6 ~]# virsh define /etc/xen/xen100.fenestros.loc.xml
Domaine xen100.fenestros.loc défini depuis /etc/xen/xen100.fenestros.loc.xml
```

list

La sous-commande **list** nous permet de générer une liste des machines virtuelles :

```
[root@centos6 ~]# virsh list
ID Nom                  État
-----
 0 Domain-0             en cours d'exécution
 1 xen100.fenestros.loc idle
```

Notez qu'avec l'option **-all**, nous pouvons également voir les machines arrêtées :

shutdown et destroy

Pour éteindre des machines il convient d'utiliser la sous-commande **shutdown** ou **destroy** :

```
[root@centos6 ~]# virsh shutdown xen100.fenestros.loc
Le domaine xen100.fenestros.loc est en cours d'arrêt
```

```
[root@centos6 ~]# virsh destroy xen100.fenestros.loc
Domaine xen100.fenestros.loc détruit
```

start

Pour démarrer une machine, il convient d'utiliser la sous-commande **start** :

```
[root@centos6 ~]# virsh start xen100.fenestros.loc
Domaine xen100.fenestros.loc démarré
```

help

Les sous-commandes de la commande **virsh** sont nombreuses :

virsh [options] [commandes]

options :

-c --connect <uri>	URI de connexion à l'hyperviseur
-r --readonly	se connecter en lecture seule
-d --debug <num>	niveau de débogage [0-5]
-h --help	afficher cette aide
-q --quiet	mode silencieux
-t --timing	afficher les informations d'horodatage
-l --log <fichier>	envoyer les informations de journalisation dans le fichier
-v --version	afficher la version du programme

commandes (mode non interactif) :

help	imprimer l'aide
attach-device	attacher un périphérique depuis un fichier XML
attach-disk	attacher un périphérique disque
attach-interface	attacher une interface réseau
autostart	démarrer automatiquement un domaine
capabilities	capacités
connect	(re)connecter à l'hyperviseur
console	se connecter à la console invitée
create	créer un domaine depuis un fichier XML
start	démarrer un domaine (précédemment défini)
destroy	détruire un domaine
detach-device	détacher un périphérique depuis un fichier XML
detach-disk	détacher un périphérique disque
detach-interface	détacher une interface réseau
define	définir (mais ne pas démarrer) un domaine depuis un fichier XML
domid	convertir un nom de domaine ou UUID en ID de domaine
domuuid	convertir un ID ou un nom de domaine en UUID de domaine
dominfo	informations du domaine
domname	convertir l'ID ou l'UUID du domaine en nom de domaine
domstate	état du domaine
domblkstat	retourner les statistiques d'un périphérique en mode bloc pour un domaine

domifstat	obtenir les statistiques d'une interface réseau pour un domaine
dumpxml	informations du domaine en XML
edit	edit XML configuration for a domain
find-storage-pool-sources	discover potential storage pool sources
find-storage-pool-sources-as	find potential storage pool sources
freecell	Mémoire NUMA disponible
hostname	afficher le nom d'hôte de l'hyperviseur
list	lister les domaines
migrate	migrer un domaine vers un autre hôte
net-autostart	démarrer automatiquement un réseau
net-create	créer un réseau depuis un fichier XML
net-define	définir (mais ne pas démarrer) un réseau depuis un fichier XML
net-destroy	détruire un réseau
net-dumpxml	informations du réseau en XML
net-edit	edit XML configuration for a network
net-list	lister les réseaux
net-name	convertir l'UUID d'un réseau en nom de réseau
net-start	démarrer un réseau inactif (précédemment défini)
net-undefine	supprimer la définition d'un réseau inactif
net-uuid	convertir le nom d'un réseau en UUID de réseau
nodeinfo	informations du noeud
nodedev-list	enumerate devices on this host
nodedev-dumpxml	node device details in XML
nodedev-dettach	dettach node device its device driver
nodedev-reattach	reattach node device its device driver
nodedev-reset	reset node device
nodedev-create	create a device defined by an XML file on the node
nodedev-destroy	destroy a device on the node
pool-autostart	démarrer automatiquement un pool
pool-build	construire un pool
pool-create	créer un pool depuis un fichier XML
pool-create-as	créer un pool depuis un ensemble d'arguments
pool-define	définir (mais ne pas démarrer) un pool depuis un fichier XML
pool-define-as	définir un pool à partir d'un ensemble d'argument

pool-destroy	détruire un pool
pool-delete	effacer un pool
pool-dumpxml	informations du pool en XML
pool-edit	edit XML configuration for a storage pool
pool-info	informations du pool de stockage
pool-list	lister les pools
pool-name	convertir l'UUID d'un pool en nom de pool
pool-refresh	rafraichir un pool
pool-start	démarrer un pool inactif (précédemment défini)
pool-undefine	supprimer un pool inactif
pool-uuid	convertir le nom d'un pool en UUID de pool
quit	quitter ce terminal interactif
reboot	redémarrer un domaine
restore	restaurer un domaine à partir d'un état sauvé dans un fichier
resume	réactiver un domaine
save	enregistrer l'état du domaine dans un fichier
schedinfo	montrer/définir les paramètres du planificateur
dump	vider l'espace mémoire d'un domaine dans un fichier pour analyse
shutdown	arrêter un domaine proprement
setmem	changer la mémoire allouée
setmaxmem	changer la limite maximum de mémoire
setvcpus	changer le nombre de processeurs virtuels
suspend	suspendre un domaine
ttyconsole	console TTY
undefine	supprimer un domaine inactif
uri	afficher l'URI canonique de l'hyperviseur
vol-create	créer un volume depuis un fichier XML
vol-create-as	créer un volume depuis un ensemble d'arguments
vol-delete	supprimer un volume
vol-dumpxml	informations du volume en XML
vol-info	informations du volume de stockage
vol-list	lister les volumes
vol-path	convertir l'UUID d'un volume en chemin de volume
vol-name	convertir l'UUID d'un volume en nom de volume

vol-key	convertir l'UUID d'un volume en clé de volume
vcpuinfo	informations sur les VCPU du domaine
vcupin	contrôler l'affinité des VCPU du domaine
version	afficher la version
vncdisplay	affichage vnc

(spécifier help <commande> pour plus de détails sur la commande)

Heureusement il existe de l'aide pour chacune d'elles :

```
xenalive:/etc/xen# virsh help vncdisplay
NAME
  vncdisplay - vnc display

SYNOPSIS
  vncdisplay <domain>

DESCRIPTION
  Output the IP address and port number for the VNC display.

OPTIONS
  <domain>          domain name, id or uuid
```

Les Outils Graphiques

Le programme virt-viewer

Virt-viewer est un utilitaire pour afficher la console graphique d'une machine virtuelle en utilisant le protocole VNC.

```
syntax: virt-viewer [OPTIONS] DOMAIN-NAME|ID|UUID
```

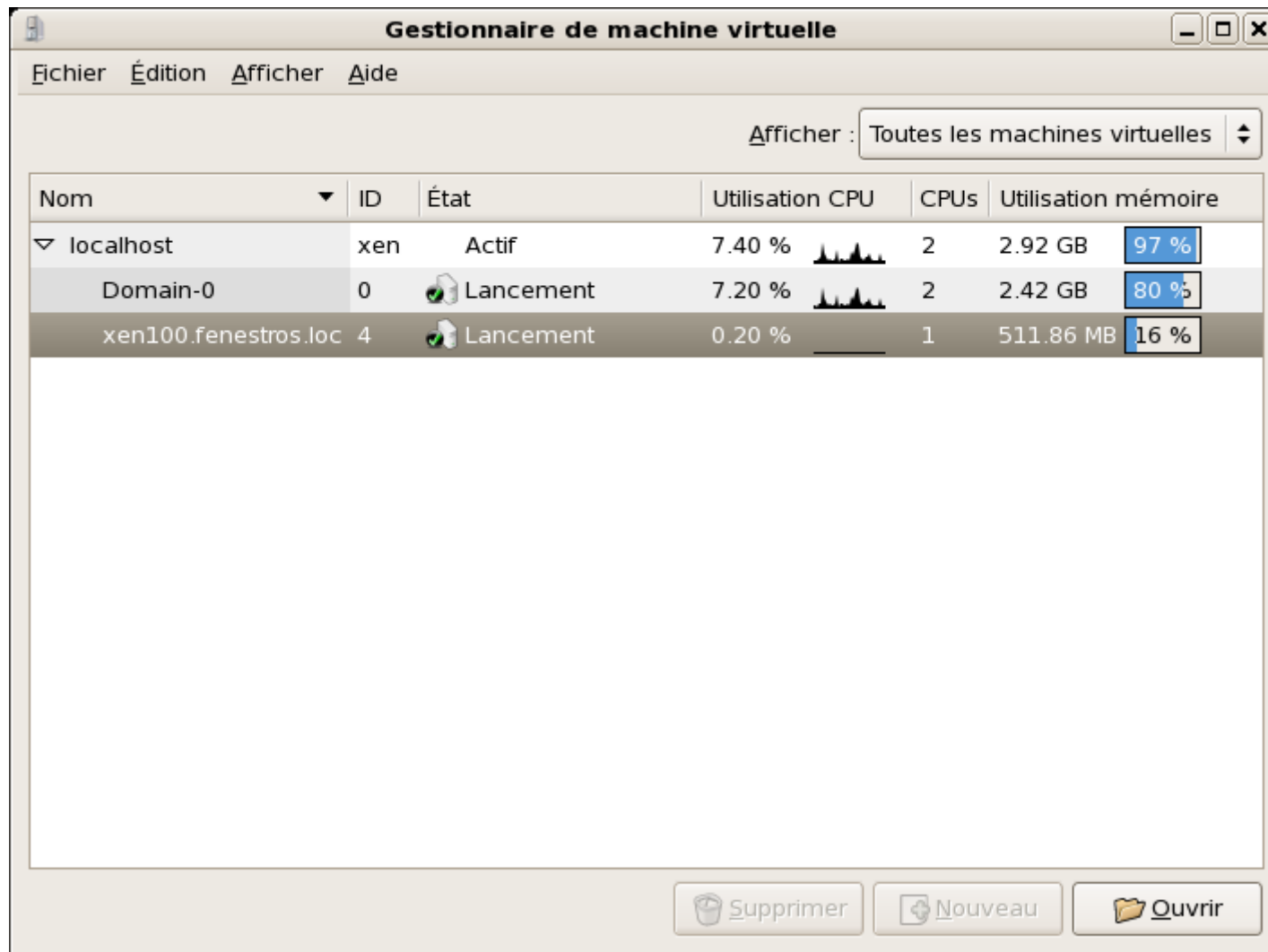
```
virt-viewer version 0.0.3
```

Options:

-h, --help	display command line help
-v, --verbose	display verbose information
-V, --version	display version information
-d, --direct	direct connection with no automatic tunnels
-c URI, --connect URI	connect to hypervisor URI
-w, --wait	wait for domain to start

Le programme virt-manager

Issu de la distribution RedHat, cet outil est l'outil standard de l'administration graphique :



L'outil ConVirt

Le dernier né des outils graphiques est ConVirt. Pour installer ConVirt, utilisez le dépôt ConVirt :

Resolving Dependencies

```
--> Running transaction check
---> Package convirt.noarch 0:1.1-1.fedora set to be updated
--> Processing Dependency: python-paramiko >= 1.6.4 for package: convirt
--> Processing Dependency: socat for package: convirt
--> Processing Dependency: vnc for package: convirt
--> Processing Dependency: tuncctl for package: convirt
--> Running transaction check
---> Package python-paramiko.noarch 0:1.7.4-1.el5.rf set to be updated
--> Processing Dependency: python-crypto for package: python-paramiko
---> Package socat.x86_64 0:1.7.1.0-1.el5.rf set to be updated
---> Package tuncctl.x86_64 0:1.5-2.el5 set to be updated
---> Package vnc.x86_64 0:4.1.2-14.el5_3.1 set to be updated
--> Running transaction check
---> Package python-crypto.x86_64 0:2.0.1-13.1.el5.kb.1 set to be updated
--> Finished Dependency Resolution
```

Dependencies Resolved

Package	Arch	Version
Repository	Size	
Installing:		
convirt	noarch	1.1-1.fedora
convirt	2.0 M	
Installing for dependencies:		
python-crypto	x86_64	2.0.1-13.1.el5.kb.1
convirt-dep	194 k	
python-paramiko	noarch	1.7.4-1.el5.rf
rpmforge	998 k	
socat	x86_64	1.7.1.0-1.el5.rf

rpmforge	398 k	
tunctl	x86_64	1.5-2.el5
convirt-dep	8.5 k	
vnc	x86_64	4.1.2-14.el5_3.1
base	158 k	

Transaction Summary

```
=====
Install      6 Package(s)
Update       0 Package(s)
Remove       0 Package(s)
```

Total download size: 3.7 M

Is this ok [y/N]: y

Downloading Packages:

(1/6): tunctl-1.5-2.el5.x86_64.rpm

| 8.5 kB 00:00

(2/6): vnc-4.1.2-14.el5_3.1.x86_64.rpm

| 158 kB 00:00

(3/6): python-crypto-2.0.1-13.1.el5.kb.1.x86_64.rpm

| 194 kB 00:01

(4/6): socat-1.7.1.0-1.el5.rf.x86_64.rpm

| 398 kB 00:01

(5/6): python-paramiko-1.7.4-1.el5.rf.noarch.rpm

| 998 kB 00:06

(6/6): convirt-1.1-1.fedora.noarch.rpm

| 2.0 MB 00:13

Total

152 kB/s | 3.7 MB 00:25

warning: rpmts_HdrFromFdno: Header V3 DSA signature: NOKEY, key ID 32a349c9

convirt/gpgkey

```
| 1.7 kB      00:00
Importing GPG key 0x32A349C9 "Convirture Corp (Key for package signing) <info@convirture.com>" from
http://www.convirture.com/repos/convirture_packaging_pub_key
Is this ok [y/N]: y
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing      : socat
1/6
  Installing      : python-crypto
2/6
  Installing      : tunctl
3/6
  Installing      : vnc
4/6
  Installing      : python-paramiko
5/6
  Installing      : convirt
6/6

Installed:
  convirt.noarch 0:1.1-1.fedora

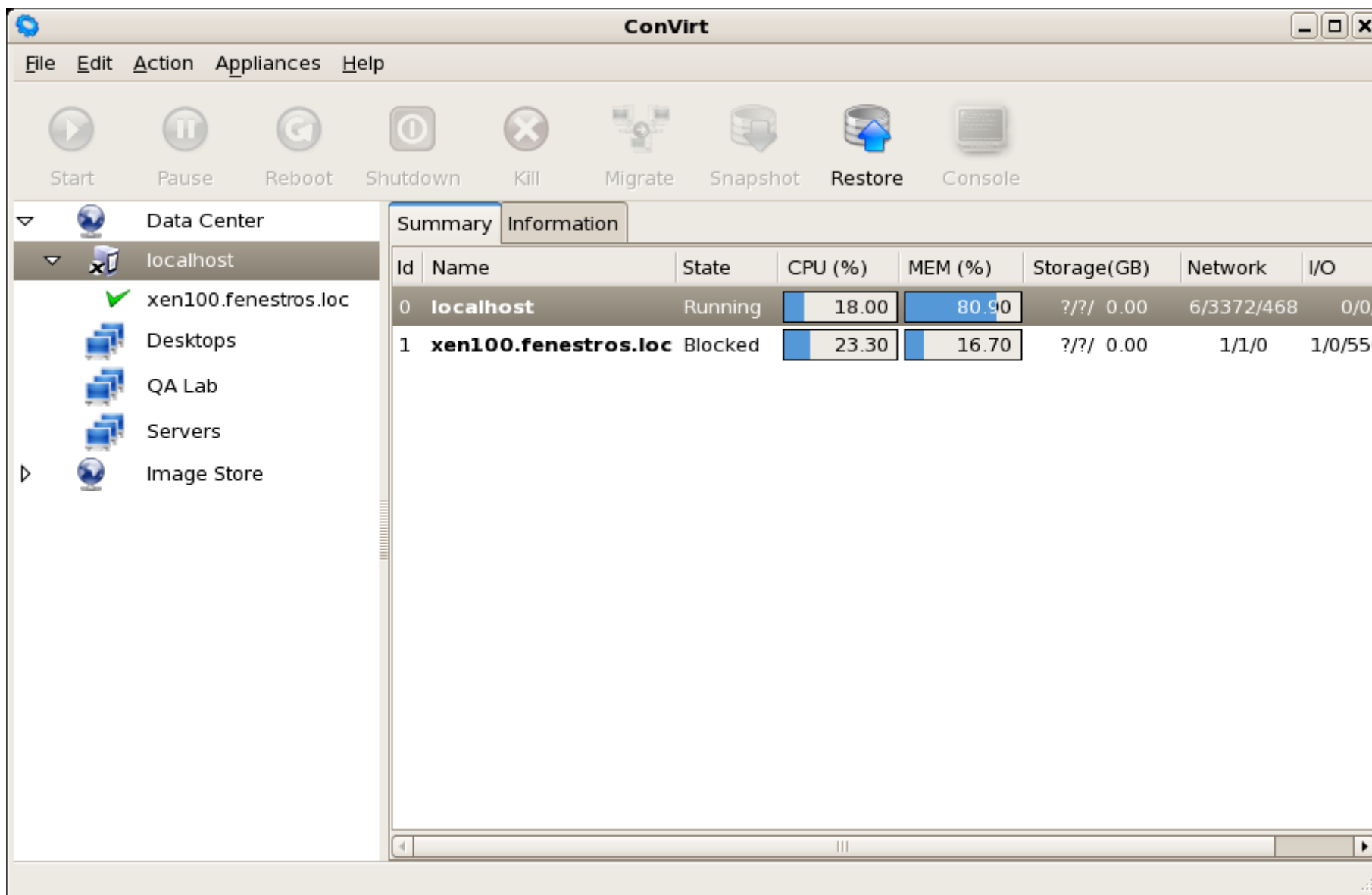
Dependency Installed:
  python-crypto.x86_64 0:2.0.1-13.1.el5.kb.1      python-paramiko.noarch 0:1.7.4-1.el5.rf      socat.x86_64
0:1.7.1.0-1.el5.rf      tunctl.x86_64 0:1.5-2.el5
  vnc.x86_64 0:4.1.2-14.el5_3.1

Complete!
```

Pour plus d'information sur ConVirt, consultez le site <http://www.convirture.com/wiki/index.php?title=Installation>.

Lancez ConVirt en ligne de commande :

```
[root@centos6 ~]# convirt &
```



The screenshot shows the ConVirt web interface. The left sidebar displays a tree structure under 'Data Center' with 'localhost' expanded, showing 'xen100.fenestros.loc' (marked with a green checkmark), 'Desktops', 'QA Lab', 'Servers', and 'Image Store'. The main panel has two tabs: 'Summary' (selected) and 'Information'. The 'Summary' tab contains a table with the following data:

Id	Name	State	CPU (%)	MEM (%)	Storage(GB)	Network	I/O
0	localhost	Running	18.00	80.90	?/?/ 0.00	6/3372/468	0/0,
1	xen100.fenestros.loc	Blocked	23.30	16.70	?/?/ 0.00	1/1/0	1/0/55



A faire - Évaluez les deux produits graphiques et comparez les options disponibles.



Important : Toutes les versions de Xen antérieures à la version Xen 4.4 utilisaient la commande **xm**. Depuis, ce **toolstack** a été remplacé par **xl**.

LAB#1 - Utilisation du nouveau toolstack (xl) sous CentOS 7

Dans le cas de Red Hat/CentOS la procédure d'installation de la distribution offre la possibilité d'installer KVM et non Xen. Xen est cependant disponible pour CentOS.

Vérifiez que votre processeur soit compatible avec la virtualisation assistée par le matériel :

```
[root@zen ~]# egrep '(vmx|svm)' /proc/cpuinfo
```

Commencez par désactiver SELINUX :

```
[trainee@zen ~]$ su -
Password:
[root@zen ~]# setenforce permissive
[root@zen ~]# vi /etc/sysconfig/selinux
[root@zen ~]# cat /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
```

```
#    disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three two values:
#    targeted - Targeted processes are protected,
#    minimum - Modification of targeted policy. Only selected processes are protected.
#    mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Installez ensuite le paquet **bridge-utils** :

```
[root@zen ~]# yum install bridge-utils
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: centos.mirrors.benatherton.com
* extras: centos.mirrors.benatherton.com
* updates: centos.mirrors.benatherton.com
Package bridge-utils-1.5-9.el7.x86_64 already installed and latest version
Nothing to do
```

Mettez en place le dépôt CentOS SIG :

```
[root@zen ~]# yum install centos-release-xen
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: centos.mirrors.benatherton.com
* extras: centos.mirrors.benatherton.com
* updates: centos.mirrors.benatherton.com
Resolving Dependencies
--> Running transaction check
---> Package centos-release-xen.x86_64 10:8-1.el7 will be installed
--> Processing Dependency: /usr/bin/grub-bootxen.sh for package: 10:centos-release-xen-8-1.el7.x86_64
--> Processing Dependency: /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-SIG-Virtualization for package: 10:centos-release-xen-8-1.el7.x86_64
--> Running transaction check
```

```
---> Package centos-release-virt-common.noarch 0:1-1.el7.centos will be installed
---> Package centos-release-xen-common.x86_64 10:8-1.el7 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

```
=====
Package                Arch      Version           Repository      Size
=====
Installing:
centos-release-xen      x86_64    10:8-1.el7        extras          6.1 k
Installing for dependencies:
centos-release-virt-common noarch    1-1.el7.centos    extras          4.5 k
centos-release-xen-common x86_64    10:8-1.el7        extras          6.7 k
=====
```

Transaction Summary

```
=====
Install 1 Package (+2 Dependent packages)
```

Total download size: 17 k

Installed size: 4.6 k

Is this ok [y/d/N]:

Mettez à jour le système :

```
[root@zen ~]# yum update
...
```

Dernièrement installez Xen :

```
[root@zen ~]# yum install xen
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: centos.mirrors.benatherton.com
```

```
* extras: centos.mirrors.benatherton.com
* updates: centos.mirrors.benatherton.com
Resolving Dependencies
--> Running transaction check
---> Package xen.x86_64 0:4.6.6-6.el7 will be installed
--> Processing Dependency: xen-runtime = 4.6.6-6.el7 for package: xen-4.6.6-6.el7.x86_64
--> Processing Dependency: libxenstore.so.3.0()(64bit) for package: xen-4.6.6-6.el7.x86_64
--> Processing Dependency: libxenguest.so.4.6()(64bit) for package: xen-4.6.6-6.el7.x86_64
--> Processing Dependency: libxenctrl.so.4.6()(64bit) for package: xen-4.6.6-6.el7.x86_64
--> Running transaction check
---> Package xen-libs.x86_64 0:4.6.6-6.el7 will be installed
--> Processing Dependency: xen-licenses for package: xen-libs-4.6.6-6.el7.x86_64
---> Package xen-runtime.x86_64 0:4.6.6-6.el7 will be installed
--> Processing Dependency: xen-hypervisor-abi = 4.6 for package: xen-runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.8.3)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.8.2)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.8.1)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.6.0)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.12.5)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.12.4)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.12.3)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.12.2)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.11.2)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.10.4)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
```

```
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.10.3)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.10.2)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.10.1)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1(SPICE_SERVER_0.10.0)(64bit) for package: xen-
runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: /usr/bin/qemu-img for package: xen-runtime-4.6.6-6.el7.x86_64
--> Processing Dependency: libspice-server.so.1()(64bit) for package: xen-runtime-4.6.6-6.el7.x86_64
--> Running transaction check
---> Package qemu-img.x86_64 10:1.5.3-141.el7_4.2 will be installed
--> Processing Dependency: libgfapi.so.0(GFAPI_3.5.0)(64bit) for package: 10:qemu-img-1.5.3-141.el7_4.2.x86_64
--> Processing Dependency: libgfapi.so.0(GFAPI_3.4.0)(64bit) for package: 10:qemu-img-1.5.3-141.el7_4.2.x86_64
--> Processing Dependency: libtcmalloc.so.4()(64bit) for package: 10:qemu-img-1.5.3-141.el7_4.2.x86_64
--> Processing Dependency: librbd.so.1()(64bit) for package: 10:qemu-img-1.5.3-141.el7_4.2.x86_64
--> Processing Dependency: libiscsi.so.2()(64bit) for package: 10:qemu-img-1.5.3-141.el7_4.2.x86_64
--> Processing Dependency: libgfapi.so.0()(64bit) for package: 10:qemu-img-1.5.3-141.el7_4.2.x86_64
---> Package spice-server.x86_64 0:0.12.8-2.el7.1 will be installed
---> Package xen-hypervisor.x86_64 0:4.6.6-6.el7 will be installed
---> Package xen-licenses.x86_64 0:4.6.6-6.el7 will be installed
--> Running transaction check
---> Package glusterfs-api.x86_64 0:3.8.4-18.4.el7.centos will be installed
---> Package gperftools-libs.x86_64 0:2.4-8.el7 will be installed
--> Processing Dependency: libunwind.so.8()(64bit) for package: gperftools-libs-2.4-8.el7.x86_64
---> Package libiscsi.x86_64 0:1.9.0-7.el7 will be installed
---> Package librbd1.x86_64 1:0.94.5-2.el7 will be installed
--> Running transaction check
---> Package libunwind.x86_64 2:1.2-2.el7 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

=====

Package	Arch	Version	Repository	Size
=====				
Installing:				
xen	x86_64	4.6.6-6.el7	centos-virt-xen	117 k
Installing for dependencies:				
glusterfs-api	x86_64	3.8.4-18.4.el7.centos	base	69 k
gperftools-libs	x86_64	2.4-8.el7	base	272 k
libiscsi	x86_64	1.9.0-7.el7	base	60 k
librbd1	x86_64	1:0.94.5-2.el7	base	1.8 M
libunwind	x86_64	2:1.2-2.el7	base	57 k
qemu-img	x86_64	10:1.5.3-141.el7_4.2	updates	677 k
spice-server	x86_64	0.12.8-2.el7.1	updates	398 k
xen-hypervisor	x86_64	4.6.6-6.el7	centos-virt-xen	925 k
xen-libs	x86_64	4.6.6-6.el7	centos-virt-xen	570 k
xen-licenses	x86_64	4.6.6-6.el7	centos-virt-xen	87 k
xen-runtime	x86_64	4.6.6-6.el7	centos-virt-xen	11 M
Transaction Summary				
=====				
Install 1 Package (+11 Dependent packages)				
Total download size: 16 M				
Installed size: 38 M				
Is this ok [y/d/N]: y				

Vérifiez que le processus d'installation de Xen a bien modifié le fichier **/boot/grub2/grub.cfg** :

```
...
### BEGIN /etc/grub.d/08_linux_xen ###
menuentry 'CentOS Linux, with Xen hypervisor' --class centos --class gnu-linux --class gnu --class os --class xen
$menuentry_id_option 'xen-gnulinux-simple-d85c6e65-858c-4552-b9da-a6927cae4f5b' {
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
```

```
if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 4a667a35-b467-4795-8557-df9710b91d1b
else
    search --no-floppy --fs-uuid --set=root 4a667a35-b467-4795-8557-df9710b91d1b
fi
echo 'Loading Xen 4.6.6-6.el7 ...'
if [ "$grub_platform" = "pc" -o "$grub_platform" = "" ]; then
    xen_rm_opts=
else
    xen_rm_opts="no-real-mode edd=off"
fi
multiboot /xen-4.6.6-6.el7.gz placeholder dom0_mem=1024M,max:1024M cpuid=0 com1=115200,8n1
console=com1,tty loglvl=all guest_loglvl=all ${xen_rm_opts}
echo 'Loading Linux 4.9.63-29.el7.x86_64 ...'
module /vmlinuz-4.9.63-29.el7.x86_64 placeholder root=UUID=d85c6e65-858c-4552-b9da-a6927cae4f5b ro rhgb
quiet console=hvc0 earlyprintk=xen nomodeset
echo 'Loading initial ramdisk ...'
module --nounzip /initramfs-4.9.63-29.el7.x86_64.img
}
...
```

Redémarrez la machine.

Connectez-vous en tant que trainee, puis passez en tant que root dans un terminal.

Pour constater le démarrage du Dom0, utilisez la commande **xl dmesg**



A faire - Implémentez toutes les étapes vues précédemment avec CentOS 6.

LAB#2 - Utilisation du nouveau toolstack (xl) sous Debian 8

Vérifiez que votre processeur soit compatible avec la virtualisation assistée par le matériel :

```
root@xen:~# egrep '(vmx|svm)' /proc/cpuinfo
```

Mettez à jour les dépôts de Debian :

```
root@xen:~# apt-get update
```

Mettez à jour l'ensemble des paquets du système :

```
root@xen:~# apt-get upgrade
```

ensuite installez le paquet **xen-system** :

```
root@xen:~# apt-get install xen-system
```

Lors de l'installation des paquets, le fichier **/boot/grub/grub.cfg** a été modifié. Afin que Xen soit lancé en premier lieu, modifiez l'ordre de démarrage des section de ce fichier :

```
root@xen:~# dpkg-divert --divert /etc/grub.d/08_linux_xen --rename /etc/grub.d/20_linux_xen
```

Mettez à jour votre fichier **/boot/grub/grub.cfg** :

```
root@xen:~# update-grub
```

Vérifiez ensuite que le fichier **/boot/grub/grub.cfg** comporte la section **/etc/grub.d/08_linux_xen** :

```
...  
### BEGIN /etc/grub.d/08_linux_xen ###  
menuentry 'Debian GNU/Linux, avec hyperviseur Xen' --class debian --class gnu-linux --class gnu --class os --
```

```
class xen $menuentry_id_option 'xen-gnulinux-simple-2836d3f4-0b75-44d4-a254-fabad6c4dab3' {
    insmod part_msdos
    insmod ext2
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-
baremetal=ahci0,msdos1 cbf4d2d3-6aa6-418d-afbe-aea71302aba4
    else
        search --no-floppy --fs-uuid --set=root cbf4d2d3-6aa6-418d-afbe-aea71302aba4
    fi
    echo    'Chargement de Xen 4.4-amd64...'
    if [ "$grub_platform" = "pc" -o "$grub_platform" = "" ]; then
        xen_rm_opts=
    else
        xen_rm_opts="no-real-mode edd=off"
    fi
    multiboot    /xen-4.4-amd64.gz placeholder    ${xen_rm_opts}
    echo    'Chargement de Linux 3.16.0-4-amd64...'
    module    /vmlinuz-3.16.0-4-amd64 placeholder root=/dev/mapper/debian--vg-root ro
initrd=/install/gtk/initrd.gz quiet
    echo    'Chargement du disque mémoire initial...'
    module    --nounzip    /initrd.img-3.16.0-4-amd64
}
...
```

Editez le fichier **/etc/default/grub** :

```
GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX="initrd=/install/gtk/initrd.gz"
GRUB_CMDLINE_XEN="dom0_mem=1024M,max:1024M dom0_max_vcpus=1 dom0_vcpus_pin"
```

Créez le fichier **/etc/xen/xend-config.sxp** :

```
(network-script network-bridge)
(vif-script vif-bridge)
(dom0-min-mem 1024)
(enable-dom0-ballooning no)
(total_available_memory 0)
(dom0-cpus 1)
(vncpasswd '')
```

Editez le fichier **/etc/default/xendomains** :

```
XENDOMAINS_SAVE=""
XENDOMAINS_RESTORE=false
XENDOMAINS_AUTO=/etc/xen/auto
XENDOMAINS_STOP_MAXWAIT=300
```

*

Mettez à jour GRUB :

```
root@debian:~# update-grub
```

Redémarrez la machine.

Connectez-vous en tant que trainee, puis passez en tant que root dans un terminal.

En utilisant donc le toolstack xl, constatez le démarrage de Dom0.

```
root@debian:~# xl list
```

Name	ID	Mem	VCPUs	State	Time(s)
Domain-0	0	1024	1	r-----	47.9

Installez les utilitaires dont on a besoin :

```
root@debian:~# apt-get install virt-viewer virt-manager xen-tools
```

Activez le réseau default :

```
root@debian:~# virsh net-list --all
```

Name	State	Autostart	Persistent
default	inactive	no	yes

```
root@debian:~# virsh net-start default
Network default started
```

```
root@debian:~# virsh net-list --all
```

Name	State	Autostart	Persistent
default	active	no	yes

```
root@debian:~# ip addr
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master xenbr0 state UP group default qlen 1000
    link/ether 00:00:00:00:00:0d brd ff:ff:ff:ff:ff:ff
3: xenbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 00:00:00:00:00:0d brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.53/24 brd 192.168.11.255 scope global xenbr0
        valid_lft forever preferred_lft forever
    inet6 fe80::200:ff:fe00:d/64 scope link
        valid_lft forever preferred_lft forever
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
```

```
link/ether 46:29:ce:73:77:6a brd ff:ff:ff:ff:ff:ff
inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
    valid_lft forever preferred_lft forever
```

Ouvrez le fichier **/etc/xen-tools/xen-tools.conf** avec vi et décommentez les deux lignes suivantes :

```
dir = /home/xen/
passwd = 1
```

Installez une VM avec la commande **xen-create-image** :

```
root@debian:~# xen-create-image --hostname ten --ip 192.168.122.10 --vcpus 1 --pygrub --dist jessie
```

Installez une VM avec la commande **virt-install** :

```
root@debian:~# mkdir /home/xen/domains/xen1.fenestros.loc && virt-install --name=xen1.fenestros.loc --ram=512 --
os-type=linux --paravirt --location=http://ftp.us.debian.org/debian/dists/stable/main/installer-amd64/ --disk
path=/home/xen/domains/xen1.fenestros.loc/disk.img,size=6 --vnc --network network=default
```

<html>

Copyright © 2020 Hugh Norris.

</html>
