

Version : **2022.01**

Dernière mise-à-jour : 2022/04/22 06:25

VIC601 - Présentation, Installation et Configuration

Contenu du Module

- **VIC601 - Présentation, Installation et Configuration**
 - Contenu du Module
 - Présentation
 - LAB #1 - Installation de KVM
 - 1.1 - Installation des Paquets Requis
 - 1.2 - Activation et Démarrage du Service libvirt
 - 1.3 - Modules du Noyau
 - LAB #2 - Configuration de KVM
 - 2.1 - Configuration du Pare-feu
 - 2.2 - Configuration du Réseau des VMs
 - 2.3 - Configuration du Stockage

Présentation

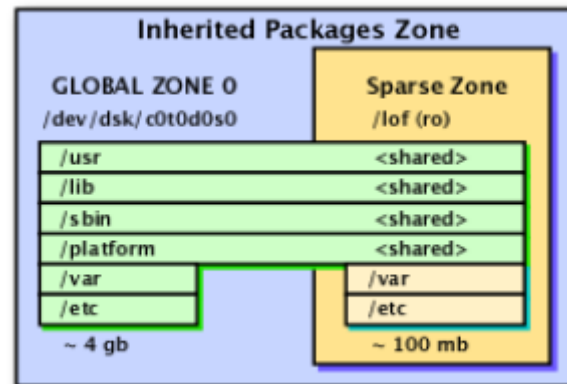
La Virtualisation

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 :

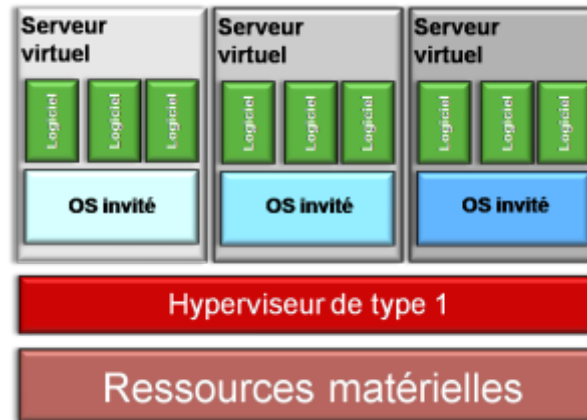
- **Virtualisation au niveau du système d'exploitation ou *L'Isolation***
-

- **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, Docker.



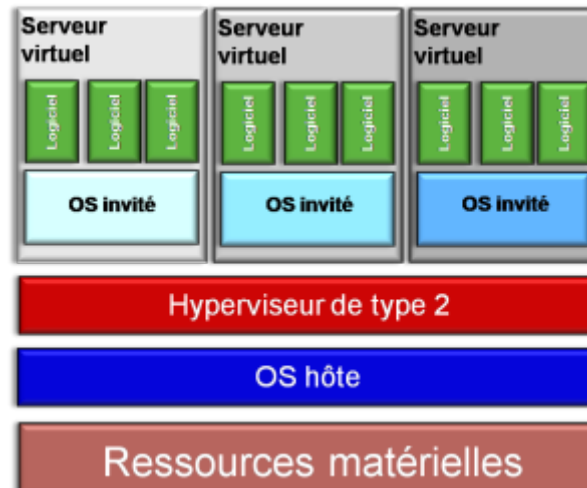
- **Paravirtualisation ou Hyperviseur de type 1**

- **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.



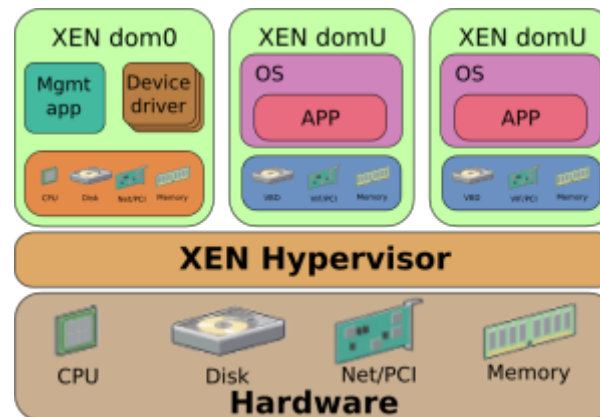
- **Virtualisation complète ou Hyperviseur de type 2**

- **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, BOCHS.



- **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

KVM

KVM ou **K**ernel based **V**irtual **M**achine :

- est un hyperviseur libre sous Linux,
- ne fonctionne que sur les architectures x86 disposant des extensions Intel-VT ou AMD-V,
- est un projet issu de QEMU.



Important : Le module KVM est intégré dans le noyau Linux depuis la version 2.6.20 et permet la paravirtualisation depuis le noyau **2.6.25**.

KVM appartient à la société **Red Hat**.

KVM permet de virtualiser :

- Windows™ à partir de Windows™ 2000,
- Toutes les distributions Linux,
- La majorité des Unix BSD,
- Solaris™ et openSolaris,
- Minux, Hurd, QNX,
- MSDOS.

KVM offre un support du matériel suivant :

- USB,
- Ethernet,
- PCI Hotplug,
- Carte Son,
- **Virtuo** - un périphérique disque paravirtualisé.

Les avantages de KVM par rapport à Xen sont :

- l'utilisation de noyaux non-modifiés au niveaux des invités,
- l'intégration direct dans le noyau Linux.

LAB #1 - Installation de KVM

1.1 - Installation des Paquets Requis

Avant d'installer KVM, il convient de savoir si le processeur de l'hôte supporte les extensions de virtualisation. Les extensions nécessaires sont soit **vmx** soit **svm** selon que le processeur du système hôte soit de la marque Intel™ ou AMD™ :

```
[root@centos8 ~]# egrep '(vmx|svm)' /proc/cpuinfo | wc -l
8
```

La majorité des paquets nécessaires pour la virtualisation sous KVM ont été regroupés dans un **module** dénommé **virt**. Il convient donc de l'installer avec la commande **dnf** :

```
[root@centos8 ~]# dnf module install virt -y
```

Si vous souhaitez utiliser l'interface graphique (non-recommandée) pour administrer les machines virtuelles, vous devez installer le paquet **virt-viewer** :

```
[root@centos8 ~]# dnf install virt-install virt-viewer -y
```

Le paquet **bridge-utils** est requis pour la gestion des bridges réseau. Installez donc celui-ci :

```
[root@centos8 ~]# dnf install epel-release -y
[root@centos8 ~]# dnf install bridge-utils -y
```

Dernièrement, les outils dont vous aurez besoin se trouvent dans les paquets **virt-top** et **libguestfs-tools** :

```
[root@centos8 ~]# dnf install virt-top libguestfs-tools -y
```

1.2 - Activation et Démarrage du Service libvirtd

Activez et démarrez le service **libvirtd** pour démarrer KVM. Notez l'utilisation de l'option **-now** qui permet de faire les deux actions en une seule ligne de commande :

```
[root@centos8 ~]# systemctl enable --now libvirtd
```

Vérifiez le statut du service avant de poursuivre :

```
[root@centos8 ~]# systemctl status libvirtd
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2021-09-01 10:19:05 EDT; 11s ago
     Docs: man:libvirtd(8)
           https://libvirt.org
  Main PID: 7502 (libvirtd)
    Tasks: 19 (limit: 32768)
   Memory: 49.3M
    CGroup: /system.slice/libvirtd.service
            └─1942 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/libexec/libvirt_leaseshelper
              └─1943 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/libexec/libvirt_leaseshelper
                └─7502 /usr/sbin/libvirtd --timeout 120

Sep 01 10:19:05 centos8.ittraining.loc systemd[1]: Starting Virtualization daemon...
Sep 01 10:19:05 centos8.ittraining.loc systemd[1]: Started Virtualization daemon.
Sep 01 10:19:06 centos8.ittraining.loc dnsmasq[1942]: read /etc/hosts - 2 addresses
Sep 01 10:19:06 centos8.ittraining.loc dnsmasq[1942]: read /var/lib/libvirt/dnsmasq/default.addnhosts - 0 addresses
```

```
Sep 01 10:19:06 centos8.ittraining.loc dnsmasq-dhcp[1942]: read /var/lib/libvirt/dnsmasq/default.hostsfile
```

1.3 - Modules du Noyau

Votre VM présente aux système hôte un processeur de type Intel™. Pour que KVM puisse fonctionner dans cet environnement il a besoin que deux modules soient chargés :

- **kvm**
- **kvm-intel**

Vérifiez donc le bon chargement des modules concernés :

```
[root@centos8 ~]# modinfo kvm
filename:      /lib/modules/4.18.0-305.7.1.el8.i2tch.x86_64/kernel/arch/x86/kvm/kvm.ko.xz
license:      GPL
author:       Qumranet
rhelversion:  8.4
srcversion:   0B52FB25C4DD9865FC4FABA
depends:      irqbypass
intree:      Y
name:        kvm
vermagic:    4.18.0-305.7.1.el8.i2tch.x86_64 SMP mod_unload modversions
sig_id:      PKCS#7
signer:      CentOS kernel signing key
sig_key:     38:77:B1:DF:46:4F:B7:5C:99:8B:B9:BF:70:A4:10:85:91:7F:50:C1
sig_hashalgo: sha256
signature:   24:2A:F9:57:2C:FB:D8:B1:3B:4B:FA:70:3B:FC:7E:23:E1:AC:06:5B:
              8D:0B:83:15:A5:56:43:04:23:FD:89:B1:BD:2C:8B:D0:0D:99:8C:9B:
              16:E5:E8:BB:C2:32:EE:62:18:E2:38:F9:2D:9D:1A:68:CA:63:3C:3B:
              B1:02:2E:46:FF:DF:6C:28:33:79:6D:FA:63:0A:B3:BA:DC:C7:FC:1D:
              FF:14:21:6C:AC:4B:67:2E:EC:A3:B8:96:A0:02:8C:BB:34:CA:CE:23:
              BD:34:10:B0:87:B6:55:7C:A9:24:1C:3A:94:9B:05:66:0D:E5:EE:28:
              CF:EE:74:9C:D3:CD:96:07:92:F6:D0:6C:FF:94:67:7A:C2:BB:26:BD:
```

```
A9:AA:04:7C:19:64:1C:3D:9D:20:EE:1F:12:C1:D2:64:D3:36:62:22:
CD:3D:F7:45:0F:E3:09:89:AC:11:54:AA:C5:56:E7:FE:CA:0C:AD:2A:
FB:60:47:5A:70:6D:AF:F2:4D:6B:60:53:EA:95:4B:E9:BB:F0:F7:71:
19:6A:60:F5:12:6D:C9:E3:86:37:AF:41:88:E3:08:47:F0:C1:C4:76:
90:FE:47:81:36:3C:CE:BD:C9:44:30:99:B8:44:3E:F6:85:B3:44:9A:
D9:76:64:92:3A:49:5A:67:91:35:B1:C7:E0:82:AF:8F:05:2C:CB:18:
50:F1:4F:4E:B0:C3:D4:AE:37:9F:1B:FF:E9:AC:4C:7B:23:65:DA:0E:
FE:98:1B:5A:D3:AB:6B:6A:EA:3C:7B:EC:54:84:CB:DB:EE:CD:F1:43:
40:F0:A3:DB:9B:0A:F2:0A:1F:59:5C:88:3C:3C:CB:CF:3D:60:54:50:
60:CC:C1:41:6A:C6:8B:7E:23:F6:D8:1B:5F:A1:73:98:D4:F8:1E:C7:
D0:F5:F0:DF:E7:2A:79:A1:E6:A1:EE:B4:69:4C:15:DC:A8:A1:40:54:
EC:9D:86:AF:B2:1D:DB:33:F8:63:5D:CD:58:12:F9:C2:FB:B6:19:EE:
7F:CC:6E:6E
```

```
parm:      tdp_mmu:bool
parm:      nx_huge_pages:bool
parm:      nx_huge_pages_recovery_ratio:uint
parm:      flush_on_reuse:bool
parm:      ignore_msrs:bool
parm:      report_ignored_msrs:bool
parm:      min_timer_period_us:uint
parm:      kvmclock_periodic_sync:bool
parm:      tsc_tolerance_ppm:uint
parm:      lapic_timer_advance_ns:int
parm:      vector_hashing:bool
parm:      enable_vmware_backdoor:bool
parm:      force_emulation_prefix:bool
parm:      pi_inject_timer:bint
parm:      halt_poll_ns:uint
parm:      halt_poll_ns_grow:uint
parm:      halt_poll_ns_grow_start:uint
parm:      halt_poll_ns_shrink:uint
```

```
[root@centos8 ~]# modinfo kvm_intel
filename:   /lib/modules/4.18.0-305.7.1.el8.i2tch.x86_64/kernel/arch/x86/kvm/kvm-intel.ko.xz
```

```
license:      GPL
author:       Qumranet
rhelversion:  8.4
srcversion:   E25F50CB67CEEDD925DE618
alias:        cpu:type:x86,ven*fam*mod*:feature:*0085*
depends:       kvm
intree:       Y
name:         kvm_intel
vermagic:     4.18.0-305.7.1.el8.i2tch.x86_64 SMP mod_unload modversions
sig_id:       PKCS#7
signer:       CentOS kernel signing key
sig_key:      38:77:B1:DF:46:4F:B7:5C:99:8B:B9:BF:70:A4:10:85:91:7F:50:C1
sig_hashalgo: sha256
signature:    7A:B3:78:ED:6D:A0:85:21:C0:5B:21:FC:7D:54:A1:04:99:38:11:3B:
              11:A3:D6:05:DF:C2:DE:46:CE:CB:86:DD:1D:0E:7F:D5:6B:44:BC:DA:
              91:EC:0C:D8:F1:BA:60:BF:B1:D5:3A:DE:A7:DA:9B:F0:16:FD:B1:37:
              BB:DC:28:39:0D:99:92:45:1D:9B:19:42:9D:F7:9F:79:BB:31:86:8C:
              E4:00:18:B8:40:85:96:D2:F3:91:11:9A:8A:CD:90:1F:2A:A6:99:3A:
              2B:B1:D7:A5:52:D5:DA:5A:C9:57:23:42:7B:36:91:F1:CD:21:DC:87:
              56:53:C9:2F:23:9E:E3:F2:C7:17:46:90:F0:A5:46:C4:C7:46:8A:09:
              3B:17:34:8F:EB:C8:42:1C:06:E8:21:AB:D5:BA:66:3E:71:9C:0E:A1:
              6C:51:7B:19:DF:02:F5:39:8A:8D:09:87:8E:F0:61:04:5E:A7:01:76:
              51:E6:7A:C7:56:D5:AF:29:02:C1:0D:13:28:C1:5C:01:D9:13:44:3C:
              D0:B9:58:0C:46:AF:ED:9E:BE:C2:70:48:35:5F:DE:77:F4:29:16:FA:
              25:E2:FC:93:A4:8A:CC:69:DC:C1:11:0B:3A:24:D0:81:A2:2A:B8:E7:
              97:4F:EB:EF:AB:75:85:63:4D:DA:C9:45:D1:AE:86:A3:B2:66:97:48:
              7D:4D:2A:59:B6:AC:F7:CF:14:2A:5B:9D:40:5A:AF:DC:62:A7:EC:55:
              AC:4D:5F:E6:C1:EA:51:2B:EF:59:30:67:91:39:C1:E9:9D:A9:70:30:
              2E:25:37:A3:F2:3E:5A:5B:98:A6:EA:75:E7:AE:42:31:62:C1:A5:6A:
              8D:CA:7F:28:A3:52:C4:65:6F:6D:BA:D1:BA:47:1A:AC:25:E0:CB:58:
              D7:27:D2:85:88:45:3D:4A:AB:39:5E:FE:42:22:43:79:B4:AC:7C:39:
              A3:5E:8B:8E:81:6B:18:DB:4F:F3:A4:D8:72:6D:97:9B:85:D2:18:35:
              E6:C7:D9:84
parm:         enable_shadow_vmcs:bool
```

```
parm:      nested_early_check:bool
parm:      vpid:bool
parm:      vnmi:bool
parm:      flexpriority:bool
parm:      ept:bool
parm:      unrestricted_guest:bool
parm:      eptad:bool
parm:      emulate_invalid_guest_state:bool
parm:      fasteoi:bool
parm:      enable_apicv:bool
parm:      nested:bint
parm:      pml:bool
parm:      dump_invalid_vmcs:bool
parm:      preemption_timer:bool
parm:      allow_smaller_maxphyaddr:bool
parm:      ple_gap:uint
parm:      ple_window:uint
parm:      ple_window_grow:uint
parm:      ple_window_shrink:uint
parm:      ple_window_max:uint
parm:      pt_mode:int
parm:      enlightened_vmcs:bool
```

LAB #2 - Configuration de KVM

2.1 - Configuration du Pare-feu

Si vous souhaitez vous connecter aux machines virtuelles créées sous KVM en utilisant le protocole VNC, vous devez ouvrir le port **5900/tcp** dans le système hôte :

```
[root@centos8 ~]# firewall-cmd --permanent --add-port=5901/tcp
```

```
success
```

```
[root@centos8 ~]# firewall-cmd --reload  
success
```

2.2 - Configuration du Réseau des VMs

Lors de l'installation de KVM un pont a été créé ayant le nom **virbr0** et l'adresse IP **192.168.122.1/24** :

- La plage des adresses IP disponible pour les machines virtuelles KVM va de **192.168.122.2/24** à **192.168.122.254/24**,
- Ce pont met en place une connectivité de type **NAT** pour les machines virtuelles,
- Une interface réseau fictive, **virbr0-nic** et appelée une esclave, a été ajoutée à ce pont principalement pour fournir une adresse MAC stable,
- Normalement au fur et au mesure que d'autres VMs soient créées, d'autres interfaces fictives seraient ajoutées, une par VM.

Les configurations ci-dessus peuvent être visualisées grâce à la commande **ip a** :

```
[root@centos8 ~]# ip a  
...  
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000  
    link/ether 52:54:00:79:02:66 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  
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group default qlen 1000  
    link/ether 52:54:00:79:02:66 brd ff:ff:ff:ff:ff:ff
```

Dans le cas où on ne souhaite pas ou on ne peut pas utiliser le pont créé par défaut, il convient de créer un autre pont. Dans notre cas, l'hôte KVM possède deux interfaces **ens18** et **ens19**. Nous allons donc dédier l'interface **ens19** au trafic réseau des machines virtuelles. Actuellement cette interface ne possède pas d'adresse IP :

```
[root@centos8 ~]# ip a show ens19  
3: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 46:de:37:c0:55:6c brd ff:ff:ff:ff:ff:ff
```

Elle n'est pas visible dans la sortie de la commande **nmcli c show** :

```
[root@centos8 ~]# nmcli c show
```

NAME	UUID	TYPE	DEVICE
ip_fixe	0f48c74d-5d16-4c37-8220-24644507b589	ethernet	ens18
virbr0	d330b8f4-d08b-4b15-93cc-45c61e26ca6a	bridge	virbr0
ens18	fc4a4d23-b15e-47a7-bcfa-b2e08f49553e	ethernet	--

Créez donc une adresse IP de 192.168.56.2/24 pour l'interface **ens19** :

```
[root@centos8 ~]# nmcli connection add con-name ip_kvm ifname ens19 type ethernet ip4 192.168.56.2/24 gw4 192.168.56.1
Connection 'ip_kvm' (afc8b175-f2cb-47b2-baca-66454058c36f) successfully added.
[root@centos8 ~]# nmcli c show
```

NAME	UUID	TYPE	DEVICE
ip_fixe	0f48c74d-5d16-4c37-8220-24644507b589	ethernet	ens18
ip_kvm	afc8b175-f2cb-47b2-baca-66454058c36f	ethernet	ens19
virbr0	d330b8f4-d08b-4b15-93cc-45c61e26ca6a	bridge	virbr0
ens18	fc4a4d23-b15e-47a7-bcfa-b2e08f49553e	ethernet	--

Utilisez la commande **ip** pour vérifier la prise en compte de la configuration :

```
[root@centos8 ~]# ip a show ens19
3: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 46:de:37:c0:55:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.2/24 brd 192.168.56.255 scope global noprefixroute ens19
        valid_lft forever preferred_lft forever
    inet6 fe80::4b01:d543:147:dd6d/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Notez que cette configuration a été stockée dans le fichier **/etc/sysconfig/network-scripts/ifcfg-ip_kvm** :

```
[root@centos8 ~]# cat /etc/sysconfig/network-scripts/ifcfg-ip_kvm
TYPE=Ethernet
```

```
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=192.168.56.2
PREFIX=24
GATEWAY=192.168.56.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ip_kvm
UUID=afc8b175-f2cb-47b2-baca-66454058c36f
DEVICE=ens19
ONBOOT=yes
```

Vérifiez maintenant que vous pouvez communiquer avec la passerelle par défaut **192.168.56.1/24** :

```
[root@centos8 ~]# ping 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.
64 bytes from 192.168.56.1: icmp_seq=1 ttl=64 time=14.6 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=64 time=0.209 ms
64 bytes from 192.168.56.1: icmp_seq=3 ttl=64 time=0.160 ms
^C
--- 192.168.56.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.160/4.990/14.601/6.796 ms
```

Pour pouvoir gérer l'esclave par le pont qui va être créé, il faut modifier le fichier **/etc/sysconfig/network-scripts/ifcfg-ip_kvm** :

```
[root@centos8 ~]# vi /etc/sysconfig/network-scripts/ifcfg-ip_kvm
[root@centos8 ~]# cat /etc/sysconfig/network-scripts/ifcfg-ip_kvm
```

```
TYPE=Ethernet
BOOTPROTO=none
NAME=ip_kvm
UUID=afc8b175-f2cb-47b2-baca-66454058c36f
DEVICE=ens19
ONBOOT=yes
BRIDGE=virbr0
```

Pour créer le pont, il convient de créer le fichier **/etc/sysconfig/network-scripts/ifcfg-virbr0** :

```
[root@centos8 ~]# ls -l /etc/sysconfig/network-scripts/ifcfg-virbr0
ls: cannot access '/etc/sysconfig/network-scripts/ifcfg-virbr0': No such file or directory

[root@centos8 ~]# vi /etc/sysconfig/network-scripts/ifcfg-virbr0
[root@centos8 ~]# cat /etc/sysconfig/network-scripts/ifcfg-virbr0
TYPE=BRIDGE
DEVICE=virbr0
BOOTPROTO=non
ONBOOT=yes
IPADDR=192.168.56.2
NETMASK=255.255.255.0
GATEWAY=192.168.56.1
```



Important : Notez que le nom du pont est identique au pont existant. Ceci n'a pas d'importance. Notez aussi que l'adresse IP, le masque du réseau ainsi que la passerelle sont configurés dans le fichier décrivant le pont.

Pour que la configuration puisse fonctionner, il est nécessaire d'activer le routage entre les interfaces dans l'hôte KVM :

```
[root@centos8 ~]# echo net.ipv4.ip_forward = 1 >> /usr/lib/sysctl.d/60-libvirtd.conf
[root@centos8 ~]# cat /usr/lib/sysctl.d/60-libvirtd.conf
```

```
# The kernel allocates aio memory on demand, and this number limits the
# number of parallel aio requests; the only drawback of a larger limit is
# that a malicious guest could issue parallel requests to cause the kernel
# to set aside memory. Set this number at least as large as
# 128 * (number of virtual disks on the host)
# Libvirt uses a default of 1M requests to allow 8k disks, with at most
# 64M of kernel memory if all disks hit an aio request at the same time.
fs.aio-max-nr = 1048576
net.ipv4.ip_forward = 1
```

En utilisant la commande **sysctl**, appliquez la nouvelle configuration :

```
[root@centos8 ~]# /sbin/sysctl -p /usr/lib/sysctl.d/60-libvirtd.conf
fs.aio-max-nr = 1048576
net.ipv4.ip_forward = 1
```

Dernièrement, il est important de configurer le pare-feu pour le pont :

```
[root@centos8 ~]# firewall-cmd --permanent --direct --passthrough ipv4 -I FORWARD -o virbr0 -j ACCEPT
success
[root@centos8 ~]# firewall-cmd --permanent --direct --passthrough ipv4 -I FORWARD -i virbr0 -j ACCEPT
success
```

```
[root@centos8 ~]# firewall-cmd --reload
success
```

La configuration faite, vérifiez la prise en charge en utilisant la commande **ip** :

```
[root@centos8 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 4e:b1:31:bd:5d:b2 brd ff:ff:ff:ff:ff:ff
   inet 10.0.2.46/24 brd 10.0.2.255 scope global noprefixroute ens18
       valid_lft forever preferred_lft forever
   inet 192.168.1.2/24 brd 192.168.1.255 scope global noprefixroute ens18
       valid_lft forever preferred_lft forever
   inet6 fe80::5223:aeel:998e:9f27/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 46:de:37:c0:55:6c brd ff:ff:ff:ff:ff:ff
   inet 192.168.56.2/24 brd 192.168.56.255 scope global noprefixroute ens19
       valid_lft forever preferred_lft forever
   inet6 fe80::4b01:d543:147:dd6d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
   link/ether 52:54:00:79:02:66 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
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group default qlen 1000
   link/ether 52:54:00:79:02:66 brd ff:ff:ff:ff:ff:ff
```



Important : Notez que la configuration n'as **pas** été prise en compte. En effet, l'adresse IP de **virbr0** est toujours **192.168.122.1**.

Consultez maintenant la list des réseaux configurés pour KVM :

```
[root@centos8 ~]# virsh net-list
Name      State    Autostart  Persistent
-----
default   active   yes        yes
```

En utilisant la commande **virsh net-dumpxml**, il est possible de constater la configuration **actuelle** du pont :

```
[root@centos8 ~]# virsh net-dumpxml default
<network>
  <name>default</name>
  <uuid>0679ee40-befd-4f48-841e-7fc64885eb49</uuid>
  <forward mode='nat'>
    <nat>
      <port start='1024' end='65535' />
    </nat>
  </forward>
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:79:02:66' />
  <ip address='192.168.122.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.122.2' end='192.168.122.254' />
    </dhcp>
  </ip>
</network>
```

Il est donc nécessaire d'éditer cette configuration pour refléter les modifications déjà apportées en utilisant la commande **virsh net-edit** :

```
[root@centos8 ~]# virsh net-edit default
```

A l'issu de votre édition, votre fichier doit correspondre à l'exemple suivant :

```
<network>
  <name>default</name>
  <uuid>0679ee40-befd-4f48-841e-7fc64885eb49</uuid>
  <forward mode='nat' />
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:79:02:66' />
  <ip address='192.168.56.10' netmask='255.255.255.0'>
    <dhcp>
```

```
    <range start='192.168.56.11' end='192.168.56.254' />
  </dhcp>
</ip>
</network>
```

Sortez du mode édition. Vous noterez que la commande vous indique que le fichier a été modifié :

```
[ESC]:q
[root@centos8 ~]# virsh net-edit default
Network default XML configuration edited.
```

Par contre, la simple édition du fichier n'a pas modifiée la configuration en cours :

```
[root@centos8 ~]# virsh net-dumpxml default
<network>
  <name>default</name>
  <uuid>0679ee40-befd-4f48-841e-7fc64885eb49</uuid>
  <forward mode='nat'>
    <nat>
      <port start='1024' end='65535' />
    </nat>
  </forward>
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:79:02:66' />
  <ip address='192.168.122.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.122.2' end='192.168.122.254' />
    </dhcp>
  </ip>
</network>
```

Notez que même en cas de re-démarrage du service, la configuration actuelle est persistante :

```
[root@centos8 ~]# systemctl restart libvirtd
```

```
[root@centos8 ~]# systemctl status libvirtd
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-09-02 10:27:41 EDT; 7s ago
     Docs: man:libvirtd(8)
           https://libvirt.org
  Main PID: 4037 (libvirtd)
    Tasks: 19 (limit: 32768)
   Memory: 57.5M
    CGroup: /system.slice/libvirtd.service
            └─1950 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/libexec/libvirt_leaseshelper
              └─1951 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/libexec/libvirt_leaseshelper
                └─4037 /usr/sbin/libvirtd --timeout 120

Sep 02 10:27:41 centos8.ittraining.loc systemd[1]: Starting Virtualization daemon...
Sep 02 10:27:41 centos8.ittraining.loc systemd[1]: Started Virtualization daemon.
Sep 02 10:27:41 centos8.ittraining.loc dnsmasq[1950]: read /etc/hosts - 2 addresses
Sep 02 10:27:41 centos8.ittraining.loc dnsmasq[1950]: read /var/lib/libvirt/dnsmasq/default.addnhosts - 0 addresses
Sep 02 10:27:41 centos8.ittraining.loc dnsmasq-dhcp[1950]: read /var/lib/libvirt/dnsmasq/default.hostsfile

[root@centos8 ~]# virsh net-dumpxml default
<network>
  <name>default</name>
  <uuid>0679ee40-befd-4f48-841e-7fc64885eb49</uuid>
  <forward mode='nat'>
    <nat>
      <port start='1024' end='65535' />
    </nat>
  </forward>
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:79:02:66' />
</network>
```

```
<ip address='192.168.122.1' netmask='255.255.255.0'>
  <dhcp>
    <range start='192.168.122.2' end='192.168.122.254' />
  </dhcp>
</ip>
</network>
```

Re-démarrez donc votre VM :

```
[root@centos8 ~]# reboot
```

Connectez-vous de nouveau à votre VM et contrôler la sortie de la commande **nmcli c show** :

```
[root@centos8 ~]# nmcli c show
```

NAME	UUID	TYPE	DEVICE
ip_fixe	0f48c74d-5d16-4c37-8220-24644507b589	ethernet	ens18
virbr0	289e0fdd-2eb5-4dd3-811a-65a878926e56	bridge	virbr0
ip_kvm	afc8b175-f2cb-47b2-baca-66454058c36f	ethernet	ens19
ens18	fc4a4d23-b15e-47a7-bcfa-b2e08f49553e	ethernet	--

Utilisez ensuite la commande **ip** pour vérifier l'adresse IP de **virbr0** :

```
[root@centos8 ~]# ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 4e:b1:31:bd:5d:b2 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.46/24 brd 10.0.2.255 scope global noprefixroute ens18
        valid_lft forever preferred_lft forever
    inet 192.168.1.2/24 brd 192.168.1.255 scope global noprefixroute ens18
```

```

    valid_lft forever preferred_lft forever
    inet6 fe80::5223:aee1:998e:9f27/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
3: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master virbr0 state UP group default qlen 1000
    link/ether 46:de:37:c0:55:6c brd ff:ff:ff:ff:ff:ff
4: virbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 52:54:00:79:02:66 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.10/24 brd 192.168.56.255 scope global virbr0
        valid_lft forever preferred_lft forever
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:79:02:66 brd ff:ff:ff:ff:ff:ff

```

Dernièrement, vérifier la connectivité avec la passerelle **192.162.56.1/24** :

```

[root@centos8 ~]# ping 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.
64 bytes from 192.168.56.1: icmp_seq=1 ttl=64 time=14.8 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=64 time=0.154 ms
64 bytes from 192.168.56.1: icmp_seq=3 ttl=64 time=0.153 ms
^C
--- 192.168.56.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2023ms
rtt min/avg/max/mdev = 0.153/5.030/14.785/6.897 ms

```

2.3 - Configuration du Stockage

KVM a besoin d'un emplacement pour stocker les VMs créées. L'hôte KVM a été configuré avec un disque supplémentaire **sdd** d'une taille de **32 Go** à cet effet :

```

[root@centos8 ~]# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   32G  0 disk

```

```

└─sda1          8:1    0    1G    0 part /boot
└─sda2          8:2    0   31G    0 part
   └─cl_centos8-root 253:0  0 27.8G  0 lvm  /
   └─cl_centos8-swap 253:1  0  3.2G  0 lvm  [SWAP]
sdb            8:16    0    4G    0 disk
sdc            8:32    0   64G    0 disk
└─sdc1          8:33    0   64G    0 part /home
sdd            8:48    0   32G    0 disk
sr0           11:0     1 1024M    0 rom

```

Créez donc sur **/dev/sdd** un volume physique ainsi qu'un groupe de volumes appelé **kvm_storage** :

```

[root@centos8 ~]# pvcreate /dev/sdd
Physical volume "/dev/sdd" successfully created.
[root@centos8 ~]# vgcreate kvm_storage /dev/sdd
Volume group "kvm_storage" successfully created

```

```

[root@centos8 ~]# vgs
VG          #PV #LV #SN Attr   VSize   VFree
cl_centos8   1   2   0 wz--n- <31.00g    0
kvm_storage  1   0   0 wz--n- <32.00g <32.00g

```

Créez ensuite un volume logique, dénommé **kvm_lv**, occupant toute l'espace disponible dans le groupe de volumes :

```

[root@centos8 ~]# lvcreate -l +100%FREE -n kvm_lv kvm_storage
Logical volume "kvm_lv" created.
[root@centos8 ~]# lvs
LV      VG          Attr      LSize   Pool Origin Data%  Meta%   Move Log Cpy%Sync Convert
root    cl_centos8  -wi-ao---- 27.79g
swap    cl_centos8  -wi-ao----  3.20g
kvm_lv  kvm_storage -wi-a----- <32.00g

```

Créez ensuite un système de fichiers de type **xfs** sur le volume logique **kvm_lv** :

```
[root@centos8 ~]# mkfs.xfs /dev/mapper/kvm_storage-kvm_lv
meta-data=/dev/mapper/kvm_storage-kvm_lv isize=512    agcount=4, agsize=2096896 blks
         =                               sectsz=512    attr=2, projid32bit=1
         =                               crc=1        finobt=1, sparse=1, rmapbt=0
         =                               reflink=1
data      =                               bsize=4096   blocks=8387584, imaxpct=25
         =                               sunit=0      swidth=0 blks
naming    =version 2                     bsize=4096   ascii-ci=0, ftype=1
log       =internal log                  bsize=4096   blocks=4095, version=2
         =                               sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none                          extsz=4096   blocks=0, rtextents=0
Discarding blocks...Done.
```

Éditez ensuite le fichier **/etc/fstab** afin de monter automatiquement le volume logique sur **/var/lib/libvirt/images** :

```
[root@centos8 ~]# vi /etc/fstab
[root@centos8 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Wed Jun 16 06:21:32 2021
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/cl_centos8-root /                    xfs     defaults        0 0
UUID=1c04981e-5317-4b73-9695-3ce25246835d /boot        ext4     defaults        1 2
/dev/mapper/cl_centos8-swap swap         swap     defaults        0 0
UUID=f76d6b66-985b-4a91-af9c-4987e8c1443c /home        ext4     defaults,usrquota,grpquota 1 2

## KVM Guest Image Store
```

/dev/mapper/kvm_storage-kvm_lv	/var/lib/libvirt/images	xfs	defaults	0 0
--------------------------------	-------------------------	-----	----------	-----

Montez ensuite le volume logique :

```
[root@centos8 ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	1.8G	0	1.8G	0%	/dev
tmpfs	1.9G	0	1.9G	0%	/dev/shm
tmpfs	1.9G	9.5M	1.9G	1%	/run
tmpfs	1.9G	0	1.9G	0%	/sys/fs/cgroup
/dev/mapper/cl_centos8-root	28G	17G	12G	58%	/
/dev/sda1	976M	453M	457M	50%	/boot
/dev/sdc1	63G	22G	39G	36%	/home
tmpfs	374M	20K	374M	1%	/run/user/1000
tmpfs	374M	1.2M	373M	1%	/run/user/42

```
[root@centos8 ~]# mount -a
```

```
[root@centos8 ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	1.8G	0	1.8G	0%	/dev
tmpfs	1.9G	0	1.9G	0%	/dev/shm
tmpfs	1.9G	9.5M	1.9G	1%	/run
tmpfs	1.9G	0	1.9G	0%	/sys/fs/cgroup
/dev/mapper/cl_centos8-root	28G	17G	12G	58%	/
/dev/sda1	976M	453M	457M	50%	/boot
/dev/sdc1	63G	22G	39G	36%	/home
tmpfs	374M	20K	374M	1%	/run/user/1000
tmpfs	374M	1.2M	373M	1%	/run/user/42
/dev/mapper/kvm_storage-kvm_lv	32G	261M	32G	1%	/var/lib/libvirt/images

Notez que ce volume est actuellement vide :

```
[root@centos8 ~]# ls -l /var/lib/libvirt/images/
```

```
total 0
```

Utilisez la commande **virsh pool-list** pour voir les pools de stockage déjà configurés. Cette liste devrait être vide :

```
[root@centos8 ~]# virsh pool-list
Name    State    Autostart
-----
```

Définissez le pool de stockage **kvm-storagepool** dans KVM grâce à la commande **virsh pool-define-as** :

```
[root@centos8 ~]# virsh pool-define-as kvm-storagepool --type=dir --target /var/lib/libvirt/images/
Pool kvm-storagepool defined
```



Important : Consultez [cette page](#) pour connaître les types de pool utilisés par libvirt.

Consultez [cette page](#) pour connaître les types de pool utilisés par libvirt.

Démarrez maintenant le pool **kvm-storagepool** :

```
[root@centos8 ~]# virsh pool-start kvm-storagepool
Pool kvm-storagepool started
```

Contrôlez l'état du pool **kvm-storagepool** :

```
[root@centos8 ~]# virsh pool-list
Name                State    Autostart
-----
kvm-storagepool     active   no
```





Important : Notez que le pool ne sera pas démarré automatiquement parce que la valeur d'**autostart** est **no**.

Fixez la valeur d'autostart à **yes** grâce à la commande **virsh pool-autostart** :

```
[root@centos8 ~]# virsh pool-autostart kvm-storagepool
Pool kvm-storagepool marked as autostarted
```

```
[root@centos8 ~]# virsh pool-list
Name              State    Autostart
-----
kvm-storagepool   active   yes
```

Dernièrement, constatez les détails du pool configuré :

```
[root@centos8 ~]# virsh pool-list --all --details
Name              State    Autostart  Persistent  Capacity  Allocation  Available
-----
kvm-storagepool   running   yes        yes          31.98 GiB  260.61 MiB  31.73 GiB
```

```
[root@centos8 ~]# df -h /var/lib/libvirt/images/
Filesystem          Size  Used Avail Use% Mounted on
/dev/mapper/kvm_storage-kvm_lv  32G  261M   32G   1% /var/lib/libvirt/images
```