

# Virtualisation avec KVM

---

Arnaud Abdoul Aziz AMELINA.  
Présenté a AIS/AfNOG 2015

cette Présentation peut être télécharger sur :  
<http://www.ws.afnog.org/afnog2015/ssf/virtualisation>

# What is KVM?

---

- ♦ Machine virtuelle basée sur le Noyau
- ♦ Incorporée dans les version récentes des distributions GNU/Linux :
  - ♦ Red Hat, Fedora, Centos, Debian, Ubuntu, Kalli ...
- ♦ Une virtualisation Complète (supportant presque tous les SE)
- ♦ Outils d'administration :
  - `libvirt` (en ligne de commande)
  - `virt-manager` (mode graphique)
- ♦ Rapide
- ♦ Sans Hyperviseur Bare-métal

# A quoi comparer KVM ?

	VMware ESX	Xen	KVM	VirtualBox
Vendor	VMware	Citrix	Red Hat	Oracle
Target market	Server	Server	Server	Desktop
Bare metal	Yes	Yes	No	No
Full virt	Yes	Yes	Optional	Yes
Paravirt drivers	Yes	Yes	Yes	Yes
Runs on	Bare metal (built-in RHEL)	Linux, NetBSD, Solaris in dom0	Most Linux distributions	Some Linux distributions, Windows, OSX
PCI pass thru	Yes	Yes	Yes	No
Guest storage	IDE, SCSI, USB, FC, LVM, iSCSI, NFS, filesystems	IDE, SCSI, USB, FC, LVM, iSCSI, filesystems	IDE, SCSI, USB, FC, LVM, iSCSI, filesystems	iSCSI, filesystems
libvirt support	Yes	Yes	Yes	Yes
Price	Free (ESXi)	Free (Debian)	Free (CentOS)	Free

# Pourquoi KVM ?

---

- ♦ Avantages de KVM:
  - ♦ Officiellement supporté sur RHEL, CentOS, Fedora, Debian et Ubuntu
  - ♦ Léger et efficace
  - ♦ Pris en charge par libvirt
  - ♦ L'intégration étroite avec installation automatisée sur Linux,
- ♦ Inconvénients de KVM:
  - ♦ Relativement nouveau, moins de pilotes paravirtuels pour l'OS invité
  - ♦ Ne fonctionne que sur les hôtes Linux (+ version expérimentale FreeBSD)
  - ♦ Les outils graphiques sont moins bien développés

# Stockage KVM

---

- Tout ce qui est upporté par Qemu dans l'espace-utilisateur
  - Fichiers d'image disque
  - options de `virt-install`
- Fichiers d'image disque : `--disk path=<fichier-image>`
- Image ISO de Cd-rom : `--cdrom <fichier-image>`
- formats : `vvfat`, `vpc`, `vmdk`, `vdi`, `sheepdog`, `rbd`, `raw`, `host_cdrom`, `host_floppy`, `host_device` `file` `qed` `qcow2` `qcow` `parallels` `nbd` `dmg` `tftp` `ftps` `ftp` `https` `http` `cow` `cloop`
- Partitions (raw)
- Utiliser LVM pour la flexibilité
- Accueillir des périphériques physiques

# Le Réseau KVM

---

- ♦ Tout ce qui est supporté par Qemu dans l'espace utilisateur
- ♦ Le Mode automatique
- ♦ S'il y a un dispositif de pont dans l'hôte avec une interface physique affectée, qui pourrait être utiliser pour la connectivité
- ♦ Option de virt-install : `--network=bridge=<périphérique>`
- ♦ Remarque : l'équipement doit être un dispositif de pont sous Linux, exemple `br0`
- ♦ Généralement ça ne fonctionne pas avec une interface WiFi



# Installation du SE invité (démonstration)

- `sudo virt-install --connect qemu:///system \`  
`--virt-type kvm --name FreeBSD-Demo \`  
`--os-variant=freebsd8 --ram 1024 --vcpus 1 \`  
`--disk path=FreeBSD-Demo.img,size=20 \`  
`--cdrom FreeBSD-9.1-RELEASE-amd64-dvd1.iso \`  
`--network=bridge=br219 \`  
`--graphics type=vnc,listen=0.0.0.0`
- Remarque: br219 est spécifique à notre installation
  - Vous voulez probablement omettre l'option `--réseau`
- Connectez-vous à `<hôte>: 0` avec VNC pour conduire l'installation
- Démonstration en direct!





# Suppression d'une machine virtuelle (invité)

- ♦ Si vous faites une erreur avec *virt-install*
- ♦ `virsh destroy FreeBSD-Demo`
  - ♦ Stopper la VM via une extension virtuelle forcée
- ♦ `virsh undefine FreeBSD-Demo`
  - ♦ Supprimer le fichier XML de la VM depuis */etc/libvirt/qemu*
  - ♦ Impossible de démarrer ou stopper la VM
- ♦ `rm FreeBSD-Demo.img`
  - ♦ Supprimer l'image-disque de la VM
- ♦ Puis lancer *virt-install* pour en créer une autre



## Après installation du SE

---

- ♦ L'installateur éjectera le cd-rom virtuel
  - ♦ libvirt détache l'image cd-rom
  - ♦ Après le redémarrage il n'y aura plus de cd-rom image dans le lecteur virtuel
  - ♦ Comme indiqué par `virsh domblklist FreeBSD-Demo`
- ♦ L'installateur redémarrera le système
- ♦ Mais ça ne démarre pas totalement – Pourquoi?
  - ♦ Regarder dans `virsh list --all`
  - ♦ Besoin de démarrer immédiatement
  - ♦ Paramétrer l'auto-redémarrage  
`virsh autostart FreeBSD-Demo`



# Création d'une Image disque Gold

---

- ♦ Extinction propre de gold system
  - ♦ `shutdown -p now` or `poweroff`
- ♦ Vérifier si ça fonctionne
  - ♦ `virsh list -all`
- ♦ Copie les fichiers-images (Pourquoi?)
  - ♦ `sudo cp FreeBSD-Demo.img FreeBSD-Demo-Gold.img`

# Création de clone

---

```
hostname=pc$pc
macaddr=`echo $hostname | md5sum | sed -e \
's/^\(..\)..\(..\)..\(..\)..\(..\)..*/52:54:\1:\2:\3:\4/'`
image=/data/vm/$hostname.img
sudo qemu-img create -f qcow2 -o
backing_file=FreeBSD-Demo-Gold.img $image
virt-install --connect qemu:///system \
--virt-type kvm --name $hostname \
--os-variant=freebsd8 --ram 512 --vcpus 1 \
--disk path=$image,format=qcow2 \
--network=bridge=br219,mac=$macaddr \
--graphics type=vnc,listen=0.0.0.0 --import
```

Live demo!

Que dois-je changer pour en créer un autre ?

# Plus de clones!

---

```
for pc in {1..32}; do
    hostname=pc$pc
    macaddr=`echo $hostname | md5sum | sed -e 's/^\
(..)\(..)\(..)\(..)\.*/52:54:\1:\2:\3:\4/'`
    image=/data/vm/$hostname.img
    sudo qemu-img create -f qcow2 \
        -o backing_file=FreeBSD-SSE-Gold.img $image
    virt-install --connect qemu:///system \
        --virt-type kvm --name $hostname \
        --os-variant=freebsd8 --ram 512 --vcpus 1 \
        --disk path=$image,format=qcow2 \
        --network=bridge=br219,mac=$macaddr \
        --graphics type=vnc,listen=0.0.0.0 --import
done
```

# Questions

---

- ♦ De quelle taille seront les images-disque?
  - ♦ Comment leur taille évoluera-t-elle?
- ♦ Quels nom d'hôtes auront ces machines?
  - ♦ Comment le configurer automatiquement?
  - ♦ Utiliser DHCP, configurer `hostname=""` dans `/etc/rc.conf`
- ♦ Comment les gérer à la volée?
  - ♦ Comment gérer les mis-à-jour des OS?
- ♦ Quel temps machine le système utilisera?
- ♦ Qu'advient-il des journaux-système?
- ♦ Ou sont inscrites les clés SSH?

## Utilisation de DHCP pour les adresses IP fixes

```
for i in {17..32}; do
hostname=$1
ipaddr=$2

if [ -z "$ipaddr" ]; then
echo "Usage: $0 <hostname> <ip-address>" >&2
exit 2
fi

macaddr=`echo $hostname | md5sum | sed -e 's/^\
(..)\(..)\(..)\(..).* /52:54:\1:\2:\3:\4/'`
cat <<EOF
host $hostname {
hardware ethernet $macaddr;
fixed-address $ipaddr;
}
EOF

exit 0
```

# FIN

---

Questions?