Installation d'un Hyperviseur KVM sur Centos6

Table of Contents

Installation du socle

Procéder à l'installation d'un socle dgfip

Installation d'Apache

Activation des dépôts EPEL

EPEL : Extra Packages for Enterprise Linux fournit un ensemble de paquets supplémentaires pour Red

Hat Enterprise Linux (RHEL), CentOS et Scientific Linux (SL), Oracle Linux (OL).

Les paquets EPEL sont généralement basés sur leurs homologues Fedora et ne seront jamais en conflit avec ou remplacent les paquets dans les distributions de base Enterprise Linux.

Télécharger puis installer le rpm epel-release

```
wget http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
yum -y install remi-release-6.rpm epel-release-6-8.noarch.rpm
```

Activer les nouveaux dépôts

Dans /etc/yum.repos.d

Editer le fichier epel.repo pour s'assurer que la première section [epel] est activée (enablet=1):

```
 [epel]
 name=Extra Packages for Enterprise Linux 6 - $basearch
 #baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch
mirrorlist=http://mirrors.fedoraproject.org/metalink?repo=epel-6&arch=$basea
rch
 failovermethod=priority
 enabled=1
 gpgcheck=1
 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
```

Remi Collet maintient une grande collection de RPMS, y compris les dernières versions de PHP, etc. Il s'agit d'une collection de repos et l'utilisation de la série -safe garantira que rien de la base de CentOS Linux distro n'est écrasé ou remplacé. Pour installer les repos de REMI télécharger le rpm remi-release, puis installer:
wget http://rpms.famillecollet.com/enterprise/remi-release-6.rpm
yum -y install remi-release-6.rpm epel-release-6-8.noarch.rpm

Editer le fichier remi.repo pour s'assurer que la première section [remi] est activée (enabled=1):

```
[remi]
name=Les RPM de remi pour Enterprise Linux 6 - $basearch
#baseurl=http://rpms.famillecollet.com/enterprise/6/remi/$basearch/
mirrorlist=http://rpms.famillecollet.com/enterprise/6/remi/mirror
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi
```

Désactiver le cache

Dans /etc/yum.repos.d/ éditer le fichier socle-dgfip-local-rpms.repo pour désactiver le cache (enabled=0) :

```
[socle-dgfip-local-rpms]
name=socle dgfip local rpms
baseurl=file:///var/cache/master/rpms/
gpgcheck=0
enabled=0
```

Activer les dépots officiels de CEntos

Dans /etc/yum.repos.d/ éditer le fichier CentOS-Base.repo pour activer tous les repos (enabled=1) :

```
[base]
....
enabled=1
....
[updates]
....
enabled=1
....
[extras]
....
enabled=1
....
[centosplus]
....
enabled=1
....
[contrib]
....
enabled=1
....
```

Installer la clé GPG des repos Centos

```
$ wget https://centos.org/keys/RPM-GPG-KEY-CentOS-6
$ rpm --import RPM-GPG-KEY-CentOS-6
```

Faire un update

```
$ yum update
```

Installer apache et php

En fonction de la version de PHP requise il peut être nécessaire d'installer une version plus récente.

Désinstaller la version précédente

```
$ yum remove php php-common php-agent
```

Vérifier que les dépçots epel et remis soient activés

Un rechargement du cache yum doit faire apparaître les repos epel et remi

```
yum repolist
Modules complémentaires chargés : fastestmirror
Loading mirror speeds from cached hostfile
epel/metalink  21 kB 00:00
* base: centos.crazyfrogs.org
* centosplus: centos.mirror.ate.info
* contrib: centos.mirrors.ovh.net
* epel: epel.besthosting.ua
* extras: centos.crazyfrogs.org
* remi: fr2.rpmfind.net
* remi-safe: fr2.rpmfind.net
* updates: centos.mirror.ate.info
```

Installer php et ses dépendances

```
$ yum install httpd php mysql-server php-mysql pam_mysql vsftpd
```

Configuration du serveur

Activer short*open*tag dans /etc/php.ini

```
short_open_tag = on
```

Si nécessaire modifier le port d'écoute du serveur

dans /etc/httpd/conf/httpd.conf

```
Listen 8080
```

Redémarrer le service httpd

```
$ service httpd restart
```

WebVirtMgr

Installation

Installation de KVM libvirt

Installation des paquets

```
$ yum -y install kvm libvirt
```

Activation du démon libvirtd

Dé commenter dans /etc/sysconfig/libvirtd la ligne

```
LIBVIRTD_ARGS="--listen"
```

Pour les serveurs ne disposant pas d'un CA valide, décommenter la ligne lsiten_tls=0 dans /etc/libvirt/libvirtd.conf :

```
# This is enabled by default, uncomment this to disable it
listen_tls = 0
```

Démarrer le démon libvirtd

```
$ service libvirtd start
```

Ajouter les utilisateurs et leurs mots de passe avec la commande saslpasswd2. Lors de l'exécution de cette commande, il est important de préciser que le nom de l'application est libvirt. A titre d'exemple, pour ajouter un utilisateur fred :

```
$ saslpasswd2 -a libvirt fred
Password: xxxxxx
Again (for verification): xxxxxx
```

Configurer le démarrage automatique de libvirt

```
$ chkconfig libvirtd on; shutdown -r now
```

Installation de nginx

Installer les prérequis

Certains prérequis ne sont disponibles que sur les dépôts EPEL. Vérifier la présence de ceux-ci par yum update

Si EPEL n’apparaît pas activer les dépôts d'EPEL

```
 $ yum install -y git python-pip libvirt-python libxml2-python python-
websockify supervisor nginx
```

Installation de webvirtmgr

Télécharger la dernière release

```
$ git clone https://github.com/retspen/webvirtmgr.git
```

Installer les prérequis de python et paramétrer Django

```
$ cd webvirtmgr
$ pip install -r requirements.txt
$ ./manage.py syncdb
$ ./manage.py collectstatic
```

Créer un superuser

```
You just installed Django's auth system, which means you don't have any
superusers defined.
Would you like to create one now? (yes/no): yes (Put: yes)
Username (Leave blank to use 'admin'): admin (Put: your username or login)
E-mail address: username@domain.local (Put: your email)
Password: xxxxxx (Put: your password)
Password (again): xxxxxx (Put: confirm password)
Superuser created successfully.
```

Il est possible de définir des superuser supplémentaires

```
$ ./manage.py createsuperuser
```

Paramétrage de Nginx

Habituellement WebVirtMgr est uniquement disponible à partir de localhost sur le port 8000. Cette étape mettra WebVirtMgr à la disposition de tout le monde sur le port 80. Le webinterface est également protégé (pas https), ce qui signifie que tout le monde entre vous et le serveur (personnes sur le même wifi, routeur local, votre fournisseur, le fournisseur de serveurs, backbones etc.) peut voir vos informations de connexion en texte clair!

Au lieu de cela, vous pouvez également sauter cette étape complètement. En redirigeant simplement le port 8000 sur votre machine locale via SSH. Cela est beaucoup plus sûr parce WebVirtMgr ne sont pas disponibles au public plus et vous ne pouvez y accéder via une connexion cryptée.

```
ssh user@server:port -L localhost:8000:localhost:8000 -L
localhost:6080:localhost:6080
```

Déplacer le dossier webvirtmgr dans /var/www

```
$ cd ..
$ mv webvirtmgr /var/www/
$ chown -R nginx:nginx /var/www/webvirtmgr
```

Créer webvirtmgr.conf

dans /etc/nginx/conf.d:

```
server {
listen 80 default_server;
server_name $hostname;
#access_log /var/log/nginx/webvirtmgr_access_log;
location /static/ {
root /var/www/webvirtmgr/webvirtmgr; # or /srv instead of /var
expires max;
}
location / {
proxy_pass http://127.0.0.1:8000;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-for $proxy_add_x_forwarded_for;
proxy_set_header Host $host:$server_port;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_connect_timeout 600;
proxy_read_timeout 600;
proxy_send_timeout 600;
client_max_body_size 1024M; # Set higher depending on your needs
}
}
```

Invalider l'ancien bloc server

soit en commentant les lignes du bloc dans le fichier /etc/nginx/conf.d/default.conf

soit directement en modifiant l'extension du fichier default.conf

Relancer le service nginx

```
$ service nginx restart
```

Configuration

Editer le fichier /etc/supervisord.conf

Ajouter les lignes suivantes

```
 [program:webvirtmgr]
 command=/usr/bin/python /srv/www/webvirtmgr/manage.py run_gunicorn -c
/srv/www/webvirtmgr/conf/gunicorn.conf.py
 directory=/srv/www/webvirtmgr
 autostart=true
 autorestart=true
 stdout_logfile=/var/log/supervisor/webvirtmgr.log
 redirect_stderr=true
 user=nginx
 [program:webvirtmgri-console]
 command=/usr/bin/python /srv/www/webvirtmgr/console/webvirtmgr-console
 directory=/var/www/webvirtmgr
 autostart=true
 autorestart=true
 stdout_logfile=/var/log/supervisor/webvirtmgr-console.log
 redirect_stderr=true
 user=nginx
```

Redémarrer le démon

```
 $ service supervisord restart
```

Tester la connection

Avant d'ajouter l'adresse IP de votre serveur dans le centre de contrôle effectuer le test suivant

```
 virsh -c qemu+tcp://IP_address/system nodeinfo
 ------------------------------------------------------------------------
---
 Please enter your authentication name: fred
 Please enter your password: xxxxxx
 CPU model: x86_64
 CPU(s): 2
 CPU frequency: 2611 MHz
 CPU socket(s): 1
 Core(s) per socket: 2
 Thread(s) per core: 1
 NUMA cell(s): 1
 Memory size: 2019260 kB
```

Activation de l'ip forwarding

```
$ sed -i 's/^ (net.ipv4.ip_forward = ).*/1 1/' /etc/sysctl.conf; sysctl -p
```

Configurer le Firewall

Ouvrir l'accès des ports libvirt

```
 $ iptables -I INPUT -m state --state NEW -m tcp -p tcp --dport 16509 -j
ACCEPT
```

Open Vswitch

Installation du module du noyau

CentOS 6 fournit déjà un module openvswitch dans le noyau (kernel) il n'y a donc pas besoin de construire et installer openvswitch-kmod.

Pour s'en assurer taper la commande

```
 $ modinfo openvswitch
```

Qui doit retourner la version du module embarqué

```
 filename:
/lib/modules/2.6.32-642.4.2.el6.x86_64/kernel/net/openvswitch/openvswitch.ko
 license: GPL
 description: Open vSwitch switching datapath
 srcversion: 00938868C288DBF055E30F3
 depends: libcrc32c,vxlan
 vermagic: 2.6.32-642.4.2.el6.x86_64 SMP mod_unload modversions
```

Construction du Module openvswitch du noyau

Si tel n'est pas le cas il faut construire le module du noyau

Installer les outils de coonfection d'un rpm

```
 $ yum groupinstall "Development Tools" -y
 $ yum install rpmdevtools openssl-devel kernel-devel gcc redhat-rpm-config
-y
```

Créer l'utilisateur ovswitch

```
 $ adduser ovswitch
```

Télécharger openvswitch-2.3.0

```
 $ su ovswitch -c "wget
http://openvswitch.org/releases/openvswitch-2.3.0.tar.gz -O
/home/ovswitch/openvswitch-2.3.0.tar.gz"
```

```
 $ su ovswitch -c "cd ~ && tar xvfz openvswitch-2.3.0.tar.gz"
```

Préparer l'environnement rpmbuild

```
 $ su ovswitch -c "mkdir -p /home/ovswitch/rpmbuild/SOURCES"
 $ su ovswitch -c "cp /home/ovswitch/openvswitch-2.3.0.tar.gz
/home/ovswitch/rpmbuild/SOURCES/"
 $ su ovswitch -c "cp /home/ovswitch/openvswitch-2.3.0/rhel/openvswitch-
kmod.files /home/ovswitch/rpmbuild/SOURCES/"
```

Préparer le build sans le module kmod

```
 $ su ovswitch -c "sed 's/openvswitch-kmod, //g'
/home/ovswitch/openvswitch-2.3.0/rhel/openvswitch.spec >
/home/ovswitch/openvswitch-2.3.0/rhel/openvswitch_no_kmod.spec"
```

Construire le rpm

```
 $ su ovswitch -c "rpmbuild -bb
/home/ovswitch/openvswitch-2.3.0/rhel/openvswitch_no_kmod.spec"
```

Installer le module

```
 $ yum install openvswitch
```

ou

```
 $ yum localinstall -y
/home/ovswitch/rpmbuild/RPMS/x86_64/openvswitch-2.3.0-1.x86_64.rpm
```

Activer le service

```
 $ modprobe openvswitch
 $ /etc/init.d/openvswitch start
```

Installation

Installer les pré-requis

```
 $ yum groupinstall "Development Tools" -y
 $ yum install rpmdevtools openssl-devel kernel-devel gcc redhat-rpm-config
```

```
-y
```

Créer l'utilisateur ovswitch

```
 $ adduser ovswitch
```

Installer OpenVswitch

Télécharger openvswitch-2.3.0

```
 $ su ovswitch -c "wget
http://openvswitch.org/releases/openvswitch-2.3.0.tar.gz -O
/home/ovswitch/openvswitch-2.3.0.tar.gz"
 $ su ovswitch -c "cd ~ && tar xvfz openvswitch-2.3.0.tar.gz"
```

Préparer l'environnement rpmbuild

```
 $ su ovswitch -c "mkdir -p /home/ovswitch/rpmbuild/SOURCES"
 $ su ovswitch -c "cp /home/ovswitch/openvswitch-2.3.0.tar.gz
/home/ovswitch/rpmbuild/SOURCES/"
 $ su ovswitch -c "cp /home/ovswitch/openvswitch-2.3.0/rhel/openvswitch-
kmod.files /home/ovswitch/rpmbuild/SOURCES/"
```

Configuration

Créer les interfaces dans ovs

Créer les ponts

```
 $ ovs-vsctl add-br br-int
 $ ovs-vsctl add-br br-ex
```

Ajouter les interfaces physiques

```
 $ ovs-vsctl add-port br-ex eth0
```

Adapter la configuration Network du serveur

Créer l'interface ifcfg-br-ex

```
 $ vi /etc/sysconfig/network-scripts/ifcfg-br-ex
```

En y reportant IPADDR , NETMASK et GATEWAY de ifcfg-eth0

```
DEVICE=br-ex
ONBOOT=yes
BOOTPROTO=none
IPADDR=xx.xx.xxx.xxx
NETMASK=255.255.255.0
GATEWAY=xx.xx.xxx.x
```

Editer l'inetreface ifcfg-eth0

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

Retirer IPADDR, NETMASK, GATEWAY

```
DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
BOOTPROTO=none
PROMISC=yes
```

Redémarrer le service réseau

```
$ service network restart
```