Version : **2022.01**

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

VIC603 - Gestion des Clones, des Snapshots et des Sauvegardes sous KVM

Contenu du Module

- **VIC603 - Gestion des Clones, des Snapshots et des Sauvegardes sous KVM**
  - Contenu du Module
  - LAB #1 - Gestion des Clones
    - 1.1 - Création d'une VM à partir d'un Clone
  - LAB #2 - Gestion des Snapshots
    - 2.1 - Création d'un Snapshot - VM Arrêté
    - 2.2 - Création d'un Snapshot - VM en Cours d'Exécution
    - 2.3 - Restauration d'un Snapshot
    - 2.4 - Suppression de Snapshots
  - LAB #3 - Gestion des Sauvegardes
    - 3.1 - Création d'une Copie du Disque de la VM
    - 3.2 - Création d'une Copie de la Configuration de la VM
    - 3.3 - Sauvegarder les Copies

LAB #1 - Gestion des Clones

La création de clones offre une méthode rapide de création de VMs basées sur des configurations déjà existantes.

1.1 - Création d'une VM à partir d'un Clone

Commencez par vérifier l'état de la machine virtuelle **testvm1** en utilisant la commande **virsh list** :

```
[root@centos8 ~]# virsh list
 Id   Name      State
-------------------------
 3    testvm1   running
```

Important : Notez que la VM est dans un état de **running** et que son **Id** est de **3**.

Afin de procéder à la création d'un clone, arrêtez la VM avec la commande **virsh shutdown**. Cette commande peut prendre en argument l'**Id** de la VM, le **nom** de la VM ou bien l'**UUID** de la VM :

```
[root@centos8 ~]# virsh shutdown 3
Domain 3 is being shutdown
```

Vérifiez ensuite que la VM est arrêtée :

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

Utilisez la commande **virsh-clone** pour créer une deuxième VM dénommée **testvm2** ayant une image au format qcow2 stockée dans le répertoire **/var/lib/libvirt/images/** :

```
[root@centos8 ~]# virt-clone --original testvm1 --name testvm2 -f /var/lib/libvirt/images/testvm2.qcow2
Allocating 'testvm2.qcow2'
| 5.0 GB  00:00:18
```

```
Clone 'testvm2' created successfully.
```

Important : Notez la rapidité de la création du clone par rapport à la lenteur de la création d'une VM à partir d'une image ISO.

La sortie de la commande **virsh list** ne montre pas les VMs en état d'arrêt :

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

Pour voir la liste des VMs, peu importe leurs états, utilisez l'option **–all** de la commande **virsh list** :

```
[root@centos8 ~]# virsh list --all
 Id   Name      State
--------------------------
 -    testvm1   shut off
 -    testvm2   shut off
```

Le **State** peut être une des valeurs suivantes :

- **running** - la machine virtuelle est active,
- **blocked** ou **blocking** - la machine virtuelle est en attente d'E/S ou l'hyperviseur,
- **paused** - la machine virtuelle est suspendue,
- **shutdown** - la machine virtuelle est en cours d'arrêt,
- **shut off** - la machine virtuelle est éteinte et elle n'utilise **aucune** ressource système,
- **crashed** - la machine virtuelle a crashée et est arrêtée.

LAB #2 - Gestion des Snapshots

La création de snapshots offre une méthode rapide de sauvegarde des VMs.

Il existe deux types de snapshots :

- **Internal**,
  - uniquement possibles avec les images qcow2,
  - la VM est mise en pause pendant la création du snapshot,
  - ne fonctionne pas avec des pools de stockage de type LVM.
- **External**,
  - sont basés sur le concept COW (**C**opy **O**n **W**rite),
  - l'image de base est mise en mode lecture seule,
  - une image de type overlay en mode lecture/écriture et obligatoirement au format qcow2 est créée pour les écritures suivantes,
  - est compatible avec n’importe lequel des formats de l'image de base.

2.1 - Création d'un Snapshot - VM Arrêté

Utilisez la commande **virsh snapshot-list** pour consulter la liste des snapshots déjà présents pour la VM **testvm1** :

```
[root@centos8 ~]# virsh snapshot-list testvm1
 Name   Creation Time   State
-------------------------------
```

La création d'un snapshot se fait grâce à l'utilisation de la commande **virsh snapshot-create-as**. Les arguments obligatoires de cette commande sont :

- **–domain**,
  - le domaine à l'origine du snapshot,
- **–name**,
  - le nom attribué au snapshot.

```
[root@centos8 ~]# virsh snapshot-create-as --domain testvm1 --name "testvm1-snap-03092021"
Domain snapshot testvm1-snap-03092021 created
```

Vérifiez que le snapshot a bien été pris en compte :

```
[root@centos8 ~]# virsh snapshot-list testvm1
 Name                    Creation Time               State
------------------------------------------------------------
 testvm1-snap-03092021   2021-09-03 11:27:10 -0400   shutoff
```

Important : Notez que l'état est **shutoff**.

Pour obtenir des informations concernant le snapshot, il convient d'utiliser la commande **virsh snapshot-info**. L'utilisation de l'option **–current** permet d'obtenir de l'information sur le dernier snapshot effectué pour le domaine en question :

```
[root@centos8 ~]# virsh snapshot-info --domain testvm1 --current
Name:           testvm1-snap-03092021
Domain:         testvm1
Current:        yes
State:          shutoff
Location:       internal
Parent:         -
Children:       0
Descendants:    0
Metadata:       yes
```

2.2 - Création d'un Snapshot - VM en Cours d'Exécution

Il est aussi possible de prendre des snapshot d'une VM en cours d'exécution. Cependant, dans le cas de beaucoup d'activité E/S, il est toutefois recommandée d'arrêter la VM.

Démarrez donc la VM **testvm2** avec la commande **virsh start** :

```
[root@centos8 ~]# virsh start testvm2
Domain testvm2 started
```

Vérifiez ensuite l'état de la VM :

```
[root@centos8 ~]# virsh list --all
 Id   Name      State
--------------------------
 4    testvm2   running
 -    testvm1   shut off
```

Créez un snapshot de **testvm2** :

```
[root@centos8 ~]# virsh snapshot-create-as --domain testvm2 --name "testvm2-snap-03092021"
Domain snapshot testvm2-snap-03092021 created
```

Consultez la liste des snapshots pour le domaine **testvm2** :

```
[root@centos8 ~]# virsh snapshot-list testvm2
 Name                    Creation Time               State
-------------------------------------------------------------
 testvm2-snap-03092021   2021-09-03 11:29:49 -0400   running
```

Pour obtenir des informations concernant le dernier snapshot effectué pour le domaine **testvm2**, utilisez la commande suivante :

```
[root@centos8 ~]# virsh snapshot-info --domain testvm2 --current
Name:           testvm2-snap-03092021
Domain:         testvm2
Current:        yes
State:          running
Location:       internal
```

```
Parent:         -
Children:       0
Descendants:    0
Metadata:       yes
```

Important : Notez que le **State** est **running** par rapport au snapshot précédent qui avait rapporté un **State** de **shutoff**.

2.3 - Restauration d'un Snapshot

Pour restaurer un snapshot la commande virsh fournit la sous-commande **snapshot-revert**. Commencez donc par arrêter la VM :

```
[root@centos8 ~]# virsh shutdown testvm2
Domain testvm2 is being shutdown

[root@centos8 ~]# virsh list --all
 Id   Name      State
--------------------------
 -    testvm1   shut off
 -    testvm2   shut off
```

Restaurez ensuite le snapshot :

```
[root@centos8 ~]# virsh snapshot-revert --domain testvm2 --snapshotname testvm2-snap-03092021 --running

[root@centos8 ~]# virsh list --all
 Id   Name      State
--------------------------
 5    testvm2   running
```

```
 -    testvm1   shut off
```

Important : Notez que le **State** est **running** après la restauration du snapshot.

Connectez-vous à la VM grâce à la commande **virsh console** et vérifiez que celle-ci ne présente pas d'anomalies de fonctionnement :

```
[root@centos8 ~]# virsh console 5
Connected to domain testvm2
Escape character is ^]

CentOS Linux 8
Kernel 4.18.0-305.12.1.el8_4.x86_64 on an x86_64

testvm1 login: root
Mot de passe : fenestros
Dernière connexion : Fri Sep  3 16:45:59 sur ttyS0
[root@testvm1 ~]#
...
[root@testvm1 ~]# [ALT GR]+[CTRL]+[)]
[root@centos8 ~]#
```

2.4 - Suppression de Snapshots

Les snapshots peuvent être supprimés grâce à l'utilisation de la commande **virsh snapshot-delete** :

```
[root@centos8 ~]# virsh snapshot-delete --domain testvm1 --snapshotname testvm1-snap-03092021
Domain snapshot testvm1-snap-03092021 deleted

[root@centos8 ~]# virsh snapshot-delete --domain testvm2 --snapshotname testvm2-snap-03092021
Domain snapshot testvm2-snap-03092021 deleted
```

```
[root@centos8 ~]# virsh snapshot-list testvm1
 Name   Creation Time   State
-------------------------------

[root@centos8 ~]# virsh snapshot-list testvm2
 Name   Creation Time   State
-------------------------------
```

LAB #3 - Gestion des Sauvegardes

Pour effectuer une sauvegarde d'une VM en utilisant les outils classiques, il convient simplement de copier le fichier d'image ainsi que le fichier de configuration de la VM à un endroit ailleurs sur le système, voire sur un support externe.

3.1 - Création d'une Sauvegarde du Disque de la VM

Actuellement seule la VM **testvm2** est en cours d'exécution :

```
[root@centos8 ~]# virsh list
 Id   Name      State
-------------------------
 5    testvm2   running
```

Créez donc un répertoire destiné à contenir les copies créées :

```
[root@centos8 ~]# mkdir /vm-backups
```

Placez-vous dans le répertoire des images **/var/lib/libvirt/images/** :

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

```
total 4029008
-rw-------. 1 root root 5369757696 Sep  3 11:42 testvm1-os.qcow2
-rw-------. 1 qemu qemu 2261123072 Sep  4 02:25 testvm2.qcow2
```

Copiez ensuite l'image vers le répertoire **/vm-backups** :

```
[root@centos8 images]# cp -p testvm1-os.qcow2 /vm-backups/testvm1-os.qcow2-04092021 &
```

Important : Notez que cette copie est effectuée en arrière plan afin de libérer le prompt en avant plan du terminal.

3.2 - Création d'une Sauvegarde de la Configuration de la VM

La configuration de chaque VM est contenue dans un fichier au format XML dans le répertoire **/etc/libvirt/qemu/** :

```
[root@centos8 images]# cd /etc/libvirt/qemu/
[root@centos8 qemu]# ls -lR
.:
total 16
drwx------. 3 root root   42 Sep  2 10:22 networks
-rw-------. 1 root root 4936 Sep  3 10:07 testvm1.xml
-rw-------. 1 root root 4933 Sep  3 11:35 testvm2.xml

./networks:
total 4
drwx------. 2 root root  25 Aug 10 17:00 autostart
-rw-------. 1 root root 575 Sep  2 10:22 default.xml

./networks/autostart:
total 0
```

```
lrwxrwxrwx. 1 root root 14 Jun 16 06:28 default.xml -> ../default.xml
```

Si vous consultez le contenu du fichier **testvm1.xml**, vous vous apercevrez des balises contenant la configuration actuelle de la VM. Par exemple, les deux lignes suivantes démontrent que la VM a 2 Go de RAM et 1 vCPU :

```
...
  <currentMemory unit='KiB'>2097152</currentMemory>
  <vcpu placement='static'>1</vcpu>
...
```

```
[root@centos8 qemu]# cat testvm1.xml
<!--
WARNING: THIS IS AN AUTO-GENERATED FILE. CHANGES TO IT ARE LIKELY TO BE
OVERWRITTEN AND LOST. Changes to this xml configuration should be made using:
  virsh edit testvm1
or other application using the libvirt API.
-->

<domain type='kvm'>
  <name>testvm1</name>
  <uuid>d436a2df-78b0-474c-833b-3f7af8681052</uuid>
  <metadata>
    <libosinfo:libosinfo xmlns:libosinfo="http://libosinfo.org/xmlns/libvirt/domain/1.0">
      <libosinfo:os id="http://centos.org/centos/8"/>
    </libosinfo:libosinfo>
  </metadata>
  <memory unit='KiB'>2097152</memory>
  <currentMemory unit='KiB'>2097152</currentMemory>
  <vcpu placement='static'>1</vcpu>
  <os>
    <type arch='x86_64' machine='pc-q35-rhel8.2.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
```

```
    <acpi/>
    <apic/>
  </features>
  <cpu mode='host-model' check='partial'/>
  <clock offset='utc'>
    <timer name='rtc' tickpolicy='catchup'/>
    <timer name='pit' tickpolicy='delay'/>
    <timer name='hpet' present='no'/>
  </clock>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
  <pm>
    <suspend-to-mem enabled='no'/>
    <suspend-to-disk enabled='no'/>
  </pm>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2'/>
      <source file='/var/lib/libvirt/images/testvm1-os.qcow2'/>
      <target dev='vda' bus='virtio'/>
      <address type='pci' domain='0x0000' bus='0x04' slot='0x00' function='0x0'/>
    </disk>
    <disk type='file' device='cdrom'>
      <driver name='qemu' type='raw'/>
      <target dev='sda' bus='sata'/>
      <readonly/>
      <address type='drive' controller='0' bus='0' target='0' unit='0'/>
    </disk>
    <controller type='usb' index='0' model='qemu-xhci' ports='15'>
      <address type='pci' domain='0x0000' bus='0x02' slot='0x00' function='0x0'/>
    </controller>
    <controller type='sata' index='0'>
```

```
      <address type='pci' domain='0x0000' bus='0x00' slot='0x1f' function='0x2'/>
    </controller>
    <controller type='pci' index='0' model='pcie-root'/>
    <controller type='virtio-serial' index='0'>
      <address type='pci' domain='0x0000' bus='0x03' slot='0x00' function='0x0'/>
    </controller>
    <controller type='pci' index='1' model='pcie-root-port'>
      <model name='pcie-root-port'/>
      <target chassis='1' port='0x8'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x0' multifunction='on'/>
    </controller>
    <controller type='pci' index='2' model='pcie-root-port'>
      <model name='pcie-root-port'/>
      <target chassis='2' port='0x9'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
    </controller>
    <controller type='pci' index='3' model='pcie-root-port'>
      <model name='pcie-root-port'/>
      <target chassis='3' port='0xa'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2'/>
    </controller>
    <controller type='pci' index='4' model='pcie-root-port'>
      <model name='pcie-root-port'/>
      <target chassis='4' port='0xb'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x3'/>
    </controller>
    <controller type='pci' index='5' model='pcie-root-port'>
      <model name='pcie-root-port'/>
      <target chassis='5' port='0xc'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x4'/>
    </controller>
    <controller type='pci' index='6' model='pcie-root-port'>
      <model name='pcie-root-port'/>
      <target chassis='6' port='0xd'/>
```

```
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x5'/>
    </controller>
    <controller type='pci' index='7' model='pcie-root-port'>
      <model name='pcie-root-port'/>
      <target chassis='7' port='0xe'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x6'/>
    </controller>
    <interface type='bridge'>
      <mac address='52:54:00:44:af:a1'/>
      <source bridge='virbr0'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0'/>
    </interface>
    <serial type='pty'>
      <target type='isa-serial' port='0'>
        <model name='isa-serial'/>
      </target>
    </serial>
    <console type='pty'>
      <target type='serial' port='0'/>
    </console>
    <channel type='unix'>
      <target type='virtio' name='org.qemu.guest_agent.0'/>
      <address type='virtio-serial' controller='0' bus='0' port='1'/>
    </channel>
    <input type='mouse' bus='ps2'/>
    <input type='keyboard' bus='ps2'/>
    <memballoon model='virtio'>
      <address type='pci' domain='0x0000' bus='0x05' slot='0x00' function='0x0'/>
    </memballoon>
    <rng model='virtio'>
      <backend model='random'>/dev/urandom</backend>
      <address type='pci' domain='0x0000' bus='0x06' slot='0x00' function='0x0'/>
    </rng>
```

```
  </devices>
</domain>
```

Ce fichier doit donc être copié vers le répertoire **/vm-backups/** :

```
[root@centos8 qemu]# cp -p testvm1.xml /vm-backups/testvm1.xml-backup-04092021
```

Une fois le processus en arrière plan terminé, vous pouvez constater la présence des deux fichiers dans le répertoire

```
[root@centos8 vm-backups]# ls -l
total 3992904
-rw-------. 1 root root 5369757696 Sep  3 11:42 testvm1-os.qcow2-04092021
-rw-------. 1 root root       4936 Sep  3 10:07 testvm1.xml-backup-04092021
```

Notez que la taille de l'image de testvm1 est de 5 Go malgré le fait qu'il n'y ait que 1,7 Go alloué :

```
[root@centos8 ~]# virsh vol-info /var/lib/libvirt/images/testvm1-os.qcow2
Name:           testvm1-os.qcow2
Type:           file
Capacity:       5.00 GiB
Allocation:     1.70 GiB
```

3.3 - Sauvegarder les Copies des Fichiers

Ayant effectué les copies, il est maintenant possible de sauvegarder les fichiers à l'aide d'un outil classique tel **tar** :

```
[root@centos8 vm-backups]# mkdir testvm1
[root@centos8 vm-backups]# mv *04* testvm1
[root@centos8 vm-backups]# ls
testvm1

[root@centos8 vm-backups]# tar cvzf testvm1-backup-04092021.tar.gz testvm1/
```

```
testvm1/
testvm1/testvm1-os.qcow2-04092021
testvm1/testvm1.xml-backup-04092021

[root@centos8 vm-backups]# ls -l
total 846724
drwxr-xr-x. 2 root root        74 Sep  4 03:19 testvm1
-rw-r--r--. 1 root root 867042697 Sep  4 03:24 testvm1-backup-04092021.tar.gz
```

<html>

Copyright © 2022 Hugh Norris

</html>