

Version : **2022.01**

Dernière mise-à-jour : 2022/04/29 07:57

LCF607 - Gestion de KVM et des VMs

Contenu du Module

- **LCF607 - Gestion des VMs et KVM**
 - Contenu du Module
 - LAB #1 - Augmentation de la Taille du Disque
 - 1.1 - Augmenter la Taille du Disque avec la Commande qemu-img
 - Augmenter la Taille du Disque de la VM
 - Augmenter la Taille de la Partition Système de la VM
 - Augmenter la Taille du Système de Fichiers
 - 1.2 - Augmenter la Taille du Disque avec la Commande virsh
 - Augmenter la Taille du Disque de la VM
 - LAB #2 - Réparation d'un Filesystem Corrompu
 - 2.1 - Préparation
 - 2.2 - La Commande guestfish
 - LAB #3 - Gestion de la VM
 - 3.1 La Commande shutdown
 - 3.2 La Commande reboot
 - 3.3 La Commande suspend
 - 3.4 La Commande resume
 - 3.5 La Commande dominfo
 - 3.6 La Commande autostart
 - 3.7 La Commande domuuid
 - 3.8 La Commande undefine
 - 3.9 La Commande destroy
 - LAB #4 - Gestion des Ressources

- 4.1 - Augmenter les VCPUs et la Mémoire
- 4.2 - Ajouter et Supprimer un Disque Dur
- LAB #5 - Journalisation
 - 5.1 - Le Répertoire /var/log/libvirt/qemu/
 - 5.2 - Le Fichier /var/log/messages
- LAB #6 - Gestion de KVM avec Cockpit
 - 6.1 - Installation et Démarrage
 - 6.2 - Connexion à l'Interface
 - 6.3 - Visualisation de la Configuration de la VM testvm1
 - 6.4 - Démarrage de la VM testvm1
 - 6.5 - Connexion à la VM testvm1
- LAB #7 - Gestion des Machines Virtuelles avec Cockpit
 - 7.1 - Créer un Clone de la VM testvm1
 - 7.2 - Création d'une VM en Mode Graphique à partir d'un ISO
 - 7.3 - Création et Suppression d'un Snapshot de la vm testvm2
 - 7.4 - Augmentation des Ressources de la VM testvm2
 - 7.5 - Ajout et Suppression d'un Disque Dur à la VM testvm2
- LAB #8 - Gestion de KVM avec virt-manager
 - 8.1 - Installation
 - 8.2 - Connexion à l'Interface
 - 8.3 - Démarrage de la VM testvm1
 - 8.4 - Connexion à la VM testvm1
 - 8.5 - Visualisation de la Configuration de la VM testvm1
- LAB #9 - Gestion des Machines Virtuelles avec virt-manager
 - 9.1 - Créer un Clone de la VM testvm1
 - 9.2 - Création d'une VM en Mode Graphique à partir d'un ISO
 - 9.3 - Création et Suppression d'un Snapshot de la vm testvm2
 - 9.4 - Augmentation des Ressources de la VM testvm2
 - 9.5 - Ajout et Suppression d'un Disque Dur à la VM testvm2

LAB #1 - Augmentation de la Taille du Disque

1.1 - Augmenter la Taille du Disque avec la Commande `qemu-img`

Augmenter la Taille du Disque de la VM

Les images des VMs sont stockées dans le pool **kvm-storagepool** :

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

Consultez le contenu du pool **kvm-storagepool** :

```
[root@centos8 ~]# virsh vol-list kvm-storagepool
Name              Path
-----
testvm1-os.qcow2  /var/lib/libvirt/images/testvm1-os.qcow2
testvm2.qcow2     /var/lib/libvirt/images/testvm2.qcow2
```

Consultez ensuite les informations concernant l'image **testvm1-os.qcow2** :

```
[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
```

Pour pouvoir redimensionner la taille de l'image il convient d'utiliser la commande **qemu-img resize** :

```
[root@centos8 ~]# qemu-img resize /var/lib/libvirt/images/testvm1-os.qcow2 +2G
```

```
Image resized.
```

Vérifiez ensuite la prise en charge de la modification :

```
[root@centos8 ~]# virsh vol-info /var/lib/libvirt/images/testvm1-os.qcow2
Name:          testvm1-os.qcow2
Type:          file
Capacity:      7.00 GiB
Allocation:    1.70 GiB
```

Augmenter la Taille de la Partition Système de la VM

Démarrez la VM **testvm1** :

```
[root@centos8 ~]# virsh start testvm1
```

Connectez-vous à la VM grâce à la commande **virsh console** :

```
[root@centos8 ~]# virsh console testvm1
Connected to domain testvm1
Escape character is ^]
[Return]
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
```

Constatez la taille de la partition **/dev/vda3** :

```
[root@testvm1 ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINT
sr0	11:0	1	1024M	0	rom	
vda	253:0	0	7G	0	disk	
└vda1	253:1	0	1G	0	part	/boot
└vda2	253:2	0	512M	0	part	[SWAP]
└vda3	253:3	0	3,5G	0	part	/



Important : Notez que, pour le système d'exploitation de la VM, la taille du disque est toujours 5 Go (1G + 512M + 3,5G).

Constatez ensuite l'espace disponible sur **/dev/vda3** :

```
[root@testvm1 ~]# df -h
Sys. de fichiers Taille Utilisé Dispo Uti% Monté sur
devtmpfs          891M      0  891M   0% /dev
tmpfs              909M      0  909M   0% /dev/shm
tmpfs              909M   8,5M  901M   1% /run
tmpfs              909M      0  909M   0% /sys/fs/cgroup
/dev/vda3          3,5G   1,5G  2,1G  42% /
/dev/vda1          976M   153M  756M  17% /boot
tmpfs              182M      0  182M   0% /run/user/0
```

La partition **/dev/vda3** est la partition système de la VM. Pour modifier sa taille nous avons besoin d'installer les paquets **cloud-utils-growpart** et **gdisk** :

```
root@testvm1 ~]# dnf -y install cloud-utils-growpart gdisk
```

Le paquet **cloud-utils-growpart** fournit la commande **growpart** qui permet d'étendre la partition système jusqu'à la limite de l'espace disque disponible. La commande prend deux arguments :

- le nom du fichier spécial - /dev/vda,
- le numéro de la partition à augmenter en taille - **3**.

```
[root@testvm1 ~]# growpart /dev/vda 3
/usr/bin/growpart: ligne 242: 7516192768 octets, : erreur de syntaxe : opérateur arithmétique non valable (le
symbole erroné est « octets, »)
```



Important : Notez que cette commande retourne une erreur. En effet, elle ne comprend l'unité de mesure **octets**. Autrement dit, cette commande n'est compatible qu'avec des systèmes Linux en **anglais**.

Pour palier à ce problème, il convient simplement d'effacer le contenu de la variable système **\$LANG**. De cette façon, la langue par défaut sera utilisée, à savoir l'anglais américain :

```
[root@testvm1 ~]# echo $LANG
fr_FR.UTF-8
[root@testvm1 ~]# unset LANG
[root@testvm1 ~]# echo $LANG
```

En exécutant de nouveau la commande **growpart**, aucune erreur n'est apparente :

```
[root@testvm1 ~]# growpart /dev/vda 3
CHANGED: partition=3 start=3147776 old: size=7337984 end=10485760 new: size=11532255 end=14680031
```

L'utilisation de la commande **lsblk** démontre clairement l'augmentation de la taille de la partition **/dev/vda3** :

```
[root@testvm1 ~]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0      11:0    1 1024M  0 rom
vda      253:0    0    7G  0 disk
|-vda1   253:1    0    1G  0 part /boot
```

```
| -vda2 253:2    0  512M  0 part [SWAP]
`-vda3 253:3    0  5.5G  0 part /
```



Important : Notez que, pour le système d'exploitation de la VM, la taille du disque est maintenant de 7 Go (1G + 512M +5,5G).

Par contre, l'exécution de la commande **df** démontre que le système ne voit toujours un filesystem qu'une taille 5 Go :

```
[root@testvm1 ~]# df -TH
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  934M    0  934M   0% /dev
tmpfs           tmpfs     953M    0  953M   0% /dev/shm
tmpfs           tmpfs     953M  8.9M  944M   1% /run
tmpfs           tmpfs     953M    0  953M   0% /sys/fs/cgroup
/dev/vda3       xfs       3.8G  1.6G  2.2G  42% /
/dev/vda1       ext4      1.1G  161M  793M  17% /boot
tmpfs           tmpfs     191M    0  191M   0% /run/user/0
```

Augmenter la Taille du Système de Fichiers

Pour augmenter la taille du système de fichiers se trouvant sur **/dev/vda3**, il convient d'utiliser la commande **xfs_growfs** :

```
[root@testvm1 ~]# xfs_growfs /dev/vda3
meta-data=/dev/vda3          isize=512    agcount=4, agsize=229312 blks
                =             sectsz=512    attr=2, projid32bit=1
                =             crc=1        finobt=1, sparse=1, rmapbt=0
                =             reflink=1
data        =                bsize=4096    blocks=917248, imaxpct=25
                =             sunit=0      swidth=0 blks
naming      =version 2       bsize=4096  ascii-ci=0, ftype=1
```

```
log      =internal log      bsize=4096  blocks=2560, version=2
        =                  sectsz=512  sunit=0 blks, lazy-count=1
realtime =none              extsz=4096  blocks=0, rtextents=0
data blocks changed from 917248 to 1441531
```

Cette fois-ci la sortie de la commande **df** démontre bien l'augmentation de la taille du système de fichiers :

```
[root@testvm1 ~]# df -TH
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  934M    0  934M   0% /dev
tmpfs           tmpfs     953M    0  953M   0% /dev/shm
tmpfs           tmpfs     953M  8.9M  944M   1% /run
tmpfs           tmpfs     953M    0  953M   0% /sys/fs/cgroup
/dev/vda3       xfs       5.9G  1.6G  4.4G  27% /
/dev/vda1       ext4      1.1G  161M  793M  17% /boot
tmpfs           tmpfs     191M    0  191M   0% /run/user/0
[root@testvm1 ~]# [ALT GR]+[CTRL]+[ ]
[root@centos8 ~]#
```

1.2 - Augmenter la Taille du Disque avec la Commande virsh

Augmenter la Taille du Disque de la VM

La VM **testvm2** est en cours d'exécution :

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

Utilisez la commande **virsh dombklist** pour afficher l'emplacement de l'image de la VM **testvm2** :

```
[root@centos8 ~]# virsh domblklist testvm2
Target    Source
-----
vda       /var/lib/libvirt/images/testvm2.qcow2
sda       -
```

Cette fois, utilisez la commande **virsh blockresize** pour redimensionner la taille de l'image :

```
[root@centos8 ~]# virsh blockresize --domain testvm2 --path /var/lib/libvirt/images/testvm2.qcow2 --size 7G
Block device '/var/lib/libvirt/images/testvm2.qcow2' is resized
```



Important : Notez que la commande **qemu-img resize** nécessite la spécification de l'augmentation de la taille tandis que la commande **virsh blockresize** nécessite la spécification de la taille.

Vérifiez ensuite la prise en charge de la modification :

```
[root@centos8 ~]# virsh vol-info /var/lib/libvirt/images/testvm2.qcow2
Name:          testvm2.qcow2
Type:          file
Capacity:      7.00 GiB
Allocation:    2.14 GiB
```

Connectez-vous à la VM grâce à la commande **virsh console** :

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

Ré-exécutez les mêmes commandes que la dernière fois :

```
[root@testvm1 ~]# dnf -y install cloud-utils-growpart gdisk
```

```
...
```

```
[root@testvm1 ~]# unset LANG
```

```
[root@testvm1 ~]# growpart /dev/vda 3
```

```
CHANGED: partition=3 start=3147776 old: size=7337984 end=10485760 new: size=11532255 end=14680031
```

```
[root@testvm1 ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sr0	11:0	1	1024M	0	rom	
vda	253:0	0	7G	0	disk	
-vda1	253:1	0	1G	0	part	/boot
-vda2	253:2	0	512M	0	part	[SWAP]
`-vda3	253:3	0	5.5G	0	part	/

```
[root@testvm1 ~]# xfs_growfs /dev/vda3
```

meta-data=/dev/vda3	isize=512	agcount=4, agsize=229312 blks
=	sectsz=512	attr=2, projid32bit=1
=	crc=1	finobt=1, sparse=1, rmapbt=0
=	reflink=1	
data	=	bsize=4096, blocks=917248, imaxpct=25
=	sunit=0	swidth=0 blks
naming	=version 2	bsize=4096, ascii-ci=0, ftype=1
log	=internal log	bsize=4096, blocks=2560, version=2
=	sectsz=512	sunit=0 blks, lazy-count=1
realtime	=none	extsz=4096, blocks=0, rtextents=0
data blocks changed from 917248 to 1441531		

```
[root@testvm1 ~]# df -TH
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
devtmpfs	devtmpfs	934M	0	934M	0%	/dev
tmpfs	tmpfs	953M	0	953M	0%	/dev/shm
tmpfs	tmpfs	953M	8.9M	944M	1%	/run

```
tmpfs      tmpfs      953M      0    953M    0% /sys/fs/cgroup
/dev/vda3   xfs          5.9G     1.6G     4.4G    27% /
/dev/vda1   ext4         1.1G     161M     793M    17% /boot
tmpfs      tmpfs        191M      0    191M    0% /run/user/0
[root@testvm1 ~]# [ALT GR]+[CTRL]+[ ]
[root@centos8 ~]#
```

LAB #2 - Réparation d'un Filesystem Corrompu

En cas de défaillance de la procédure d'augmentation de la taille du système de fichiers de l'image, il peut être nécessaire de procéder à la réparation du celui-ci.

2.1 - Préparation

La réparation est effectuée grâce à l'utilisation de la commande **guestfish**. Cette commande est fourni par le paquet **libguestfs-tools**. Si ce paquet n'est pas déjà installé, il faut procéder à son installation :

```
[root@centos8 ~]# dnf install -y libguestfs-tools
Last metadata expiration check: 2:14:42 ago on Sat 04 Sep 2021 02:21:59 EDT.
Package libguestfs-tools-1:1.40.2-27.module_el8.4.0+783+f8734d30.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

Arrêtez la VM testvm1 :

```
[root@centos8 ~]# virsh shutdown testvm1
Domain testvm1 is being shutdown
```

2.2 - La Commande guestfish

Commencez par ajouter l'image à réparer à **guestfish** grâce à l'utilisation de l'option **-a** :

```
[root@centos8 ~]# guestfish -a /var/lib/libvirt/images/testvm1-os.qcow2
```

```
Welcome to guestfish, the guest filesystem shell for
editing virtual machine filesystems and disk images.
```

```
Type: 'help' for help on commands
      'man' to read the manual
      'quit' to quit the shell
```

```
><fs>
```

Initialisez **guestfish** avec la commande **run** :

```
[root@centos8 ~]# guestfish -a /var/lib/libvirt/images/testvm1-os.qcow2
```

```
Welcome to guestfish, the guest filesystem shell for
editing virtual machine filesystems and disk images.
```

```
Type: 'help' for help on commands
      'man' to read the manual
      'quit' to quit the shell
```

```
><fs> run
```

Listez ensuite les systèmes de fichiers de la VM :

```
[root@centos8 ~]# guestfish -a /var/lib/libvirt/images/testvm1-os.qcow2
```

```
Welcome to guestfish, the guest filesystem shell for
```

editing virtual machine filesystems and disk images.

Type: 'help' for help on commands
 'man' to read the manual
 'quit' to quit the shell

```
><fs> run
><fs> list-fileSYSTEMS
/dev/sda1: ext4
/dev/sda2: swap
/dev/sda3: xfs
```

Procédez ensuite à la réparation des systèmes de fichiers avec la commande **fsck** :

```
[root@centos8 ~]# guestfish -a /var/lib/libvirt/images/testvm1-os.qcow2
```

Welcome to guestfish, the guest filesystem shell for
editing virtual machine filesystems and disk images.

Type: 'help' for help on commands
 'man' to read the manual
 'quit' to quit the shell

```
><fs> run
><fs> lisf-fileSYSTEMS
lisf-fileSYSTEMS: unknown command
><fs> list-fileSYSTEMS
/dev/sda1: ext4
/dev/sda2: swap
/dev/sda3: xfs
><fs> fsck xfs /dev/sda3
0
><fs> fsck ext /dev/sda3
```

0



Important : Notez que le code retour de **0** indique qu'aucune erreur n'a été trouvée. Dans le cas contraire, fdisk tentera de réparer le système de fichiers.

Quittez **guestfish** à l'aide de la touche **q** :

```
[root@centos8 ~]# guestfish -a /var/lib/libvirt/images/testvm1-os.qcow2
```

```
Welcome to guestfish, the guest filesystem shell for
editing virtual machine filesystems and disk images.
```

```
Type: 'help' for help on commands
      'man' to read the manual
      'quit' to quit the shell
```

```
><fs> run
><fs> lisf-filesystems
lisf-filesystems: unknown command
><fs> list-filesystems
/dev/sda1: ext4
/dev/sda2: swap
/dev/sda3: xfs
><fs> fsck xfs /dev/sda3
0
><fs> fsck ext /dev/sda3
0
><fs> q
```

Après la réparation du système de fichiers, démarrez le VM **testvm1** :

```
[root@centos8 ~]# virsh start testvm1
Domain testvm1 started

[root@centos8 ~]#
```

LAB #3 - Gestion de la VM

3.1 La Commande shutdown

Comme il a déjà été expliqué, pour arrêter une VM, il convient d'utiliser la commande **virsh shutdown** :

```
[root@centos8 ~]# virsh shutdown testvm1
Domain testvm1 is being shutdown

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

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

3.2 La Commande reboot

La commande **virsh reboot** permet de re-démarrer une VM :

```
[root@centos8 ~]# virsh reboot 5
```

Domain 5 is being rebooted

```
[root@centos8 ~]# virsh list --all
```

Id	Name	State

5	testvm2	running
-	testvm1	shut off

3.3 La Commande suspend

La commande **virsh suspend** permet de mettre en pause une VM. Bien évidemment, une VM ne peut être mise en pause que quand celle-ci est en cours de fonctionnement :

```
[root@centos8 ~]# virsh suspend testvm1
error: Failed to suspend domain testvm1
error: Requested operation is not valid: domain is not running
```

```
[root@centos8 ~]# virsh start testvm1
Domain testvm1 started
```

```
[root@centos8 ~]# virsh list --all
Id    Name      State
-----
5     testvm2   running
11    testvm1   running
```

```
[root@centos8 ~]# virsh suspend testvm1
Domain testvm1 suspended
```

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

```
11    testvm1    paused
```

3.4 La Commande resume

La commande **virsh resume** permet d'annuler l'effet de la commande **virsh suspend** :

```
[root@centos8 ~]# virsh resume testvm1
Domain testvm1 resumed
```

```
[root@centos8 ~]# virsh list --all
 Id   Name      State
-----
  5    testvm2   running
 11    testvm1   running
```

3.5 La Commande dominfo

La commande **virsh dominfo** permet de consulter les informations relatives à un domaine donné :

```
[root@centos8 ~]# virsh dominfo 11
Id:          11
Name:        testvm1
UUID:        d436a2df-78b0-474c-833b-3f7af8681052
OS Type:     hvm
State:       running
CPU(s):      1
CPU time:    28.7s
Max memory:  2097152 KiB
Used memory: 2097152 KiB
Persistent:  yes
Autostart:   disable
```

```
Managed save:   no
Security model:  selinux
Security DOI:    0
Security label:  system_u:system_r:svirt_t:s0:c31,c201 (permissive)
```

3.6 La Commande autostart

La commande **virsh autostart** permet de configurer la VM de façon à ce que celle-ci démarre automatiquement :

```
[root@centos8 ~]# virsh autostart d436a2df-78b0-474c-833b-3f7af8681052
Domain d436a2df-78b0-474c-833b-3f7af8681052 marked as autostarted

[root@centos8 ~]# virsh dominfo 11
Id:          11
Name:        testvm1
UUID:        d436a2df-78b0-474c-833b-3f7af8681052
OS Type:     hvm
State:       running
CPU(s):      1
CPU time:    29.0s
Max memory:  2097152 KiB
Used memory: 2097152 KiB
Persistent:  yes
Autostart:   enable
Managed save: no
Security model: selinux
Security DOI: 0
Security label: system_u:system_r:svirt_t:s0:c31,c201 (permissive)
```

Pour annuler l'effet de la commande précédente, il convient d'utiliser l'option **-disable** :

```
[root@centos8 ~]# virsh autostart --disable 11
Domain 11 unmarked as autostarted
```

```
[root@centos8 ~]# virsh dominfo 11
Id:          11
Name:        testvm1
UUID:        d436a2df-78b0-474c-833b-3f7af8681052
OS Type:     hvm
State:        running
CPU(s):       1
CPU time:     29.2s
Max memory:   2097152 KiB
Used memory:  2097152 KiB
Persistent:   yes
Autostart:    disable
Managed save: no
Security model: selinux
Security DOI:  0
Security label: system_u:system_r:svirt_t:s0:c31,c201 (permissive)
```

3.7 La Commande domuuid

La commande **virsh domuuid** permet d'obtenir l'UUID du domaine :

```
[root@centos8 ~]# virsh domuuid testvm1
d436a2df-78b0-474c-833b-3f7af8681052
```

3.8 La Commande undefine

La commande **virsh undefine** permet de supprimer la configuration d'une VM :

```
[root@centos8 ~]# virsh undefine 5
Domain 5 has been undefined
```

```
[root@centos8 ~]# virsh list --all
 Id   Name      State
-----
  5    testvm2   running
 11    testvm1   running
```



Important : Notez que si la VM est en cours d'exécution au moment de l'exécution de la commande, la VM est mise en mode **transient**. La suppression effective de la configuration a lieu quand la VM est arrêtée.

3.9 La Commande destroy

La commande **virsh destroy** permet de supprimer une VM. Cette commande est l'équivalente d'un **init 0**. Par conséquent il est conseillé d'ajouter l'option **-graceful** de façon à nettoyer le cache de l'image disque avant l'arrêt :

```
[root@centos8 ~]# virsh list --all
 Id   Name      State
-----
  5    testvm2   running
 11    testvm1   running

[root@centos8 ~]# virsh destroy 5
Domain 5 destroyed

[root@centos8 ~]# virsh list --all
 Id   Name      State
-----
 11    testvm1   running

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

```
total 3.9G
-rw-----. 1 qemu qemu 5.1G Sep  4 06:21 testvm1-os.qcow2
-rw-----. 1 root root 2.2G Sep  4 06:18 testvm2.qcow2

[root@centos8 ~]# du -sh /var/lib/libvirt/images/*
1.8G    /var/lib/libvirt/images/testvm1-os.qcow2
2.2G    /var/lib/libvirt/images/testvm2.qcow2
```

LAB #4 - Gestion des Ressources

4.1 - Augmenter les VCPUs et la Mémoire

La façon de modifier les ressources d'une VM en ligne de commande est d'éditer le fichier de configuration de la VM. Actuellement, la VM **testvm1** a **2097152 KiB** de mémoire et **1** vCPU :

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

[root@centos8 ~]# virsh dominfo 11
Id:          11
Name:        testvm1
UUID:        d436a2df-78b0-474c-833b-3f7af8681052
OS Type:     hvm
State:       running
CPU(s):      1
CPU time:    36.5s
Max memory:  2097152 KiB
Used memory: 2097152 KiB
Persistent:  yes
```

```
Autostart:      disable
Managed save:  no
Security model: selinux
Security DOI:   0
Security label: system_u:system_r:svirt_t:s0:c31,c201 (permissive)
```

Comme détaillé précédemment, ces informations se trouvent dans le fichier de configuration au format XML de la VM :

```
[root@centos8 ~]# cat /etc/libvirt/qemu/testvm1.xml
...
<memory unit='KiB'>2097152</memory>
<currentMemory unit='KiB'>2097152</currentMemory>
<vcpu placement='static'>1</vcpu>
...
```

Pour modifier la configuration, il convient d'arrêter la VM :

```
[root@centos8 ~]# virsh shutdown 11
Domain 11 is being shutdown
```

Editez ensuite les valeurs suivantes dans le fichier de configuration en utilisant la commande **virsh edit** :

```
[root@centos8 ~]# virsh edit testvm1
...
<currentMemory unit='KiB'>4194304</currentMemory>
<vcpu placement='static'>4</vcpu>
...
```

Lors de votre sauvegarde du fichier, le système vous indiquera que le fichier a été modifié :

```
[root@centos8 ~]# virsh edit testvm1
Domain testvm1 XML configuration edited.
```

L'application de l'augmentation du nombre de vCPUs est immédiate. Par contre ceci n'est pas le cas pour la mémoire.

Consultez la valeur de la mémoire dans le fichier **/etc/libvirt/qemu/testvm1.xml** :

```
...
<memory unit='KiB'>4194304</memory>
<currentMemory unit='KiB'>2097152</currentMemory>
...
```

Démarrez maintenant la VM **testvm1** :

```
[root@centos8 ~]# virsh start testvm1
Domain testvm1 started

[root@centos8 ~]# virsh list --all
 Id   Name      State
-----
  1   testvm1   running

[root@centos8 ~]# virsh dominfo 1
Id:          1
Name:        testvm1
UUID:        d436a2df-78b0-474c-833b-3f7af8681052
OS Type:     hvm
State:       running
CPU(s):      4
CPU time:    41.6s
Max memory:  4194304 KiB
Used memory: 2097152 KiB
Persistent:  yes
Autostart:   disable
Managed save: no
Security model: selinux
Security DOI: 0
```

```
Security label: system_u:system_r:svirt_t:s0:c242,c376 (permissive)
```

Pour accorder à la VM le droit d'utiliser la totalité de la mémoire qui lui a été réservée, il est nécessaire d'utiliser la commande **virsh setmem** :

```
[root@centos8 ~]# virsh setmem testvm1 4194304
```

Vérifiez ensuite la prise en compte de l'augmentation de la mémoire :

```
[root@centos8 ~]# virsh dominfo 1
Id:          1
Name:        testvm1
UUID:        d436a2df-78b0-474c-833b-3f7af8681052
OS Type:     hvm
State:       running
CPU(s):      4
CPU time:    47.5s
Max memory:  4194304 KiB
Used memory: 4194304 KiB
Persistent:  yes
Autostart:   disable
Managed save: no
Security model: selinux
Security DOI: 0
Security label: system_u:system_r:svirt_t:s0:c242,c376 (permissive)
```

4.2 - Ajouter et Supprimer un Disque Dur

Pour ajouter un disque à une VM, il convient d'abord de créer ce disque. Placez-vous donc dans le répertoire **cd /var/lib/libvirt/images/** :

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

Créez un disque d'une taille de 10 Go au format **raw** :

```
[root@centos8 images]# qemu-img create -f raw testvm1-disk2.img 10G
Formatting 'testvm1-disk2.img', fmt=raw size=10737418240
```

Attachez ensuite ce disque à la VM **testvm1** :

```
[root@centos8 images]# virsh attach-disk testvm1 --source /var/lib/libvirt/images/testvm1-disk2.img --target vdb
--persistent
Disk attached successfully
```

Connectez-vous à la VM avec la commande **virsh console** :

```
[root@centos8 images]# virsh console testvm1
Connected to domain testvm1
Escape character is ^]
```

Contrôlez la présence du disque ajouté :

```
[root@testvm1 ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0        11:0    1 1024M  0 rom
vda        253:0    0    7G  0 disk
├─vda1     253:1    0    1G  0 part /boot
├─vda2     253:2    0  512M  0 part [SWAP]
└─vda3     253:3    0   5,5G  0 part /
vdb        253:16   0   10G  0 disk
```

Créez une partition sur le disque :

```
[root@testvm1 ~]# fdisk /dev/vdb
```

```
Bienvenue dans fdisk (util-linux 2.32.1).
Les modifications resteront en mémoire jusqu'à écriture.
Soyez prudent avant d'utiliser la commande d'écriture.
```

Le périphérique ne contient pas de table de partitions reconnue.
Création d'une nouvelle étiquette pour disque de type DOS avec identifiant de disque 0xcde64a28.

Commande (m pour l'aide) : n

Type de partition

- p primaire (0 primaire, 0 étendue, 4 libre)
- e étendue (conteneur pour partitions logiques)

Sélectionnez (p par défaut) :

Utilisation de la réponse p par défaut.

Numéro de partition (1-4, 1 par défaut) :

Premier secteur (2048-20971519, 2048 par défaut) :

Dernier secteur, +secteurs ou +taille{K,M,G,T,P} (2048-20971519, 20971519 par défaut) :

Une nouvelle partition 1 de type « Linux » et de taille 10 GiB a été créée.

Commande (m pour l'aide) : w

La table de partitions a été altérée.

Appel d'ioctl() pour relire la table de partitions.

[1518.252446] vdb: vdb1

Synchronisation des disques.

Créez un système de fichiers de type **xfs** sur la partition :

```
[root@testvm1 ~]# mkfs.ext4 /dev/vdb1
```

```
mke2fs 1.45.6 (20-Mar-2020)
```

```
Rejet des blocs de périphérique : complété
```

```
En train de créer un système de fichiers avec 2621184 4k blocs et 655360 i-noeuds.
```

```
UUID de système de fichiers=805d3a53-4f8f-43f1-8b2a-bf1c493f33ee
```

```
Superblocs de secours stockés sur les blocs :
```

```
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
```

```
Allocation des tables de groupe : complété
```

```
Écriture des tables d'i-noeuds : complété
```

```
Création du journal (16384 blocs) : complété  
Écriture des superblocs et de l'information de comptabilité du système de  
fichiers : complété
```

Montez le disque et vérifiez que vous pouvez y écrire :

```
[root@testvm1 ~]# mount /dev/vdb1 /mnt  
[ 1581.199551] EXT4-fs (vdb1): mounted filesystem with ordered data mode. Opts: (null)  
[root@testvm1 ~]# cd /mnt  
[root@testvm1 mnt]# touch test  
[root@testvm1 mnt]# ls  
lost+found  test
```

Détachez-vous de la VM :

```
[root@testvm1 ~]# [ALT GR]+[CTRL]+[  
[root@centos8 images]#
```

Détachez ensuite le disque **vdb** de la VM :

```
[root@centos8 images]# virsh detach-disk testvm1 vdb  
Disk detached successfully
```

Reconnectez-vous à la VM et exécutez la commande **lsblk**. Notez que celle-ci génère des erreurs dues au fait que le disque n'a pas été démonté avant d'avoir été détaché :

```
[root@testvm1 mnt]# lsblk  
[ 1879.523849] EXT4-fs error (device vdb1): ext4_find_entry:1446: inode #2: comm lsblk: reading directory lblock  
0  
[ 1879.536586] EXT4-fs error (device vdb1): ext4_find_entry:1446: inode #2: comm lsblk: reading directory lblock  
0  
[ 1879.550137] EXT4-fs error (device vdb1): ext4_find_entry:1446: inode #2: comm lsblk: reading directory lblock  
0  
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
```

```
sr0      11:0      1 1024M  0 rom
vda      253:0      0    7G   0 disk
├─vda1   253:1      0    1G   0 part /boot
├─vda2   253:2      0   512M  0 part [SWAP]
└─vda3   253:3      0   5,5G  0 part /
```

Démontez donc le disque :

```
[root@testvm1 mnt]# cd ..
[root@testvm1 /]# umount /mnt
[ 1921.363728] Buffer I/O error on dev vdb1, logical block 1081344, lost sync page write
[ 1921.372593] JBD2: Error -5 detected when updating journal superblock for vdb1-8.
[ 1921.381098] Aborting journal on device vdb1-8.
[ 1921.386369] Buffer I/O error on dev vdb1, logical block 1081344, lost sync page write
[ 1921.395319] JBD2: Error -5 detected when updating journal superblock for vdb1-8.
```

Exécutez de nouveau la commande **lsblk**. Vous verrez qu'il n'y a plus d'erreurs :

```
[root@testvm1 /]# lsblk
NAME     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0       11:0    1 1024M  0 rom
vda       253:0    0    7G   0 disk
├─vda1    253:1    0    1G   0 part /boot
├─vda2    253:2    0   512M  0 part [SWAP]
└─vda3    253:3    0   5,5G  0 part /
[root@testvm1 /]# [ALT GR]+[CTRL]+[ ]
[root@centos8 images]#
```

LAB #5 - Journalisation

5.1 - Le Répertoire `/var/log/libvirt/qemu/`

Le répertoire `/var/log/libvirt/qemu/` contient les journaux des VMs KVM :

```
[root@centos8 images]# cd /var/log/libvirt/qemu/
[root@centos8 qemu]# ls -l
total 56
-rw-----. 1 root root  4037 Sep  4 04:51 guestfs-7w7bnjy7ro65665z.log
-rw-----. 1 root root  4005 Sep  4 04:40 guestfs-xt170t6ii8uce35y.log
-rw-----. 1 root root 37531 Sep  4 07:04 testvm1.log
-rw-----. 1 root root  7179 Sep  4 06:18 testvm2.log

[root@centos8 qemu]# more testvm1.log
2021-09-03 12:32:14.206+0000: starting up libvirt version: 6.0.0, package: 35.1.
module_el8.4.0+885+5e18b468 (CentOS Buildsys <bugs@centos.org>, 2021-08-10-20:56
:57, ), qemu version: 4.2.0qemu-kvm-4.2.0-48.module_el8.4.0+783+f8734d30, kernel
: 4.18.0-305.7.1.el8.i2tch.x86_64, hostname: centos8.ittraining.loc
LC_ALL=C \
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin \
HOME=/var/lib/libvirt/qemu/domain-1-testvm1 \
XDG_DATA_HOME=/var/lib/libvirt/qemu/domain-1-testvm1/.local/share \
XDG_CACHE_HOME=/var/lib/libvirt/qemu/domain-1-testvm1/.cache \
XDG_CONFIG_HOME=/var/lib/libvirt/qemu/domain-1-testvm1/.config \
QEMU_AUDIO_DRV=none \
/usr/libexec/qemu-kvm \
-name guest=testvm1,debug-threads=on \
-S \
-object secret,id=masterKey0,format=raw,file=/var/lib/libvirt/qemu/domain-1-test
vm1/master-key.aes \
-machine pc-q35-rhel8.2.0,accel=kvm,usb=off,dump-guest-core=off \
-cpu Broadwell-IBRS,vme=on,ss=on,vmx=on,f16c=on,rdrand=on,hypervisor=on,arat=on,
tsc-adjust=on,umip=on,md-clear=on,stibp=on,arch-capabilities=on,ssbd=on,xsaveopt
=on,pdpe1gb=on,abm=on,ibpb=on,ibrs=on,amd-stibp=on,amd-ssbd=on,skip-lldfl-vmentr
```

```
y=on,pschange-mc-no=on \  
-m 3072 \  
-overcommit mem-lock=off \  
--More-- (3%)  
  
[root@centos8 qemu]# cat guestfs-7w7bnjy7ro65665z.log  
2021-09-04 08:47:50.774+0000: starting up libvirt version: 6.0.0, package: 35.1.module_el8.4.0+885+5e18b468  
(CentOS Buildsys <bugs@centos.org>, 2021-08-10-20:56:57, ), qemu version: 4.2.0qemu-  
kvm-4.2.0-48.module_el8.4.0+783+f8734d30, kernel: 4.18.0-305.7.1.el8.i2tch.x86_64, hostname:  
centos8.ittraining.loc  
LC_ALL=C \  
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin \  
HOME=/var/lib/libvirt/qemu/domain-9-guestfs-7w7bnjy7ro65 \  
XDG_DATA_HOME=/var/lib/libvirt/qemu/domain-9-guestfs-7w7bnjy7ro65/.local/share \  
XDG_CACHE_HOME=/var/lib/libvirt/qemu/domain-9-guestfs-7w7bnjy7ro65/.cache \  
XDG_CONFIG_HOME=/var/lib/libvirt/qemu/domain-9-guestfs-7w7bnjy7ro65/.config \  
QEMU_AUDIO_DRV=none \  
TMPDIR=/var/tmp \  
/usr/libexec/qemu-kvm \  
-name guest=guestfs-7w7bnjy7ro65665z,debug-threads=on \  
-S \  
-object secret,id=masterKey0,format=raw,file=/var/lib/libvirt/qemu/domain-9-guestfs-7w7bnjy7ro65/master-key.aes \  
-machine pc-i440fx-rhel7.6.0,accel=kvm,usb=off,dump-guest-core=off \  
-cpu host \  
-m 1280 \  
-overcommit mem-lock=off \  
-smp 1,sockets=1,cores=1,threads=1 \  
-uuid e6a0f63c-b33e-4ba0-bf03-9e4b9c9dd6a2 \  
-display none \  
-no-user-config \  
-nodefaults \  
-chardev socket,id=charmonitor,fd=39,server,nowait \  
-mon chardev=charmonitor,id=monitor,mode=control \  
-rtc base=utc,driftfix=slew
```

```
-global kvm-pit.lost_tick_policy=delay \
-no-hpet \
-no-reboot \
-no-acpi \
-boot strict=on \
-kernel /var/tmp/.guestfs-0/appliance.d/kernel \
-initrd /var/tmp/.guestfs-0/appliance.d/initrd \
-append 'panic=1 console=ttyS0 edd=off udevtimeout=6000 udev.event-timeout=6000 no_timer_check printk.time=1
cgroup_disable=memory usbcore.nousb cryptomgr.notests tsc=reliable 8250.nr_uarts=1 root=/dev/sdb selinux=0 quiet
TERM=xterm-256color' \
-device virtio-scsi-pci,id=scsi0,bus=pci.0,addr=0x2 \
-device virtio-serial-pci,id=virtio-serial0,bus=pci.0,addr=0x3 \
-blockdev '{"driver":"file","filename":"/var/lib/libvirt/images/testvm1-os.qcow2","node-name":"libvirt-2-
storage","cache":{"direct":false,"no-flush":false},"auto-read-only":true,"discard":"unmap"}' \
-blockdev '{"node-name":"libvirt-2-format","read-only":false,"cache":{"direct":false,"no-
flush":false},"driver":"qcow2","file":"libvirt-2-storage","backing":null}' \
-device scsi-hd,bus=scsi0.0,channel=0,scsi-id=0,lun=0,device_id=drive-scsi0-0-0-0,drive=libvirt-2-
format,id=scsi0-0-0-0,bootindex=1,write-cache=on \
-blockdev '{"driver":"file","filename":"/var/tmp/.guestfs-0/appliance.d/root","node-name":"libvirt-3-
storage","cache":{"direct":false,"no-flush":true},"auto-read-only":true,"discard":"unmap"}' \
-blockdev '{"node-name":"libvirt-3-format","read-only":true,"cache":{"direct":false,"no-
flush":true},"driver":"raw","file":"libvirt-3-storage"}' \
-blockdev '{"driver":"file","filename":"/tmp/libguestfsxtKWVr/overlay1.qcow2","node-name":"libvirt-1-
storage","cache":{"direct":false,"no-flush":true},"auto-read-only":true,"discard":"unmap"}' \
-blockdev '{"node-name":"libvirt-1-format","read-only":false,"cache":{"direct":false,"no-
flush":true},"driver":"qcow2","file":"libvirt-1-storage","backing":"libvirt-3-format"}' \
-device scsi-hd,bus=scsi0.0,channel=0,scsi-id=1,lun=0,device_id=drive-scsi0-0-1-0,drive=libvirt-1-
format,id=scsi0-0-1-0,write-cache=on \
-chardev socket,id=charserial0,path=/tmp/libguestfsNzMNMmr/console.sock \
-device isa-serial,chardev=charserial0,id=serial0 \
-chardev socket,id=charchannel0,path=/tmp/libguestfsNzMNMmr/guestfsd.sock \
-device virtserialport,bus=virtio-serial0.0,nr=1,chardev=charchannel0,id=channel0,name=org.libguestfs.channel.0 \
-object rng-random,id=objrng0,filename=/dev/urandom \
-device virtio-rng-pci,rng=objrng0,id=rng0,bus=pci.0,addr=0x4 \
```

```
-sandbox on,obsolete=deny,elevateprivileges=deny,spawn=deny,resourcecontrol=deny \  
-msg timestamp=on  
2021-09-04 08:47:50.775+0000: Domain id=9 is tainted: custom-argv  
2021-09-04 08:47:50.775+0000: Domain id=9 is tainted: host-cpu  
2021-09-04T08:51:54.913380Z qemu-kvm: terminating on signal 15 from pid 7495 (/usr/sbin/libvirtd)  
2021-09-04 08:51:55.113+0000: shutting down, reason=destroyed
```

5.2 - Le Fichier /var/log/messages

Dans le fichier **/var/log/messages** sont consignés les messages de KVM :

```
[root@centos8 qemu]# cat /var/log/messages | grep kvm | more  
Sep  1 05:47:57 centos8 kernel: kvm-clock: Using msrs 4b564d01 and 4b564d00  
Sep  1 05:47:57 centos8 kernel: kvm-clock: cpu 0, msr b3801001, primary cpu cloc  
k  
Sep  1 05:47:57 centos8 kernel: kvm-clock: using sched offset of 369654777578829  
1 cycles  
Sep  1 05:47:57 centos8 kernel: clocksource: kvm-clock: mask: 0xffffffffffffffff  
max_cycles: 0x1cd42e4dffb, max_idle_ns: 881590591483 ns  
Sep  1 05:47:57 centos8 kernel: kvm-guest: stealtime: cpu 0, msr 13bc2c080  
Sep  1 05:47:57 centos8 kernel: kvm-guest: PV spinlocks enabled  
Sep  1 05:47:57 centos8 kernel: kvm-clock: cpu 1, msr b3801041, secondary cpu cl  
ock  
Sep  1 05:47:57 centos8 kernel: kvm-guest: stealtime: cpu 1, msr 13bc6c080  
Sep  1 05:47:57 centos8 kernel: kvm-clock: cpu 2, msr b3801081, secondary cpu cl  
ock  
Sep  1 05:47:57 centos8 kernel: kvm-guest: stealtime: cpu 2, msr 13bcac080  
Sep  1 05:47:57 centos8 kernel: kvm-clock: cpu 3, msr b38010c1, secondary cpu cl  
ock  
Sep  1 05:47:57 centos8 kernel: kvm-guest: stealtime: cpu 3, msr 13bcec080  
Sep  1 05:47:57 centos8 kernel: kvm-clock: cpu 4, msr b3801101, secondary cpu cl  
ock  
Sep  1 05:47:57 centos8 kernel: kvm-guest: stealtime: cpu 4, msr 13bd2c080
```

```
Sep  1 05:47:57 centos8 kernel: kvm-clock: cpu 5, msr b3801141, secondary cpu cl
ock
--More--
```

LAB #6 - Gestion de KVM avec Cockpit

6.1 - Installation et Démarrage

Cockpit est la nouvelle interface de gestion du serveur de RHEL8 / CentOS8. Celui-ci est installé par défaut. Par contre, le module supplémentaire pour gérer des machines virtuelles ne l'est pas. Installez donc le paquet **cockpit-machines** :

```
[root@centos8 ~]# dnf install cockpit-machines
Last metadata expiration check: 0:34:02 ago on Thu 09 Sep 2021 01:47:52 EDT.
Dependencies resolved.
=====
Package                Arch      Version                               Repository    Size
=====
Installing:
cockpit-machines       noarch    238.2-1.el8                          appstream     735 k
Installing dependencies:
libvirt-dbus           x86_64    1.3.0-2.module_el8.4.0+547+a85d02ba  appstream     90 k

Transaction Summary
=====
Install 2 Packages

Total download size: 825 k
Installed size: 995 k
Is this ok [y/N]: y
```

Activez-et démarrez le socket **cockpit** :

```
[root@centos8 ~]# systemctl start --now cockpit.socket
[root@centos8 ~]# systemctl status cockpit.socket
● cockpit.socket - Cockpit Web Service Socket
   Loaded: loaded (/usr/lib/systemd/system/cockpit.socket; disabled; vendor preset: d>
   Active: active (listening) since Thu 2021-09-09 02:24:42 EDT; 9s ago
     Docs: man:cockpit-ws(8)
    Listen: [::]:9090 (Stream)
   Process: 19235 ExecStartPost=/bin/ln -snf active.motd /run/cockpit/motd (code=exite>
   Process: 19227 ExecStartPost=/usr/share/cockpit/motd/update-motd localhost (code=e>
     Tasks: 0 (limit: 100949)
    Memory: 632.0K
    CGroup: /system.slice/cockpit.socket

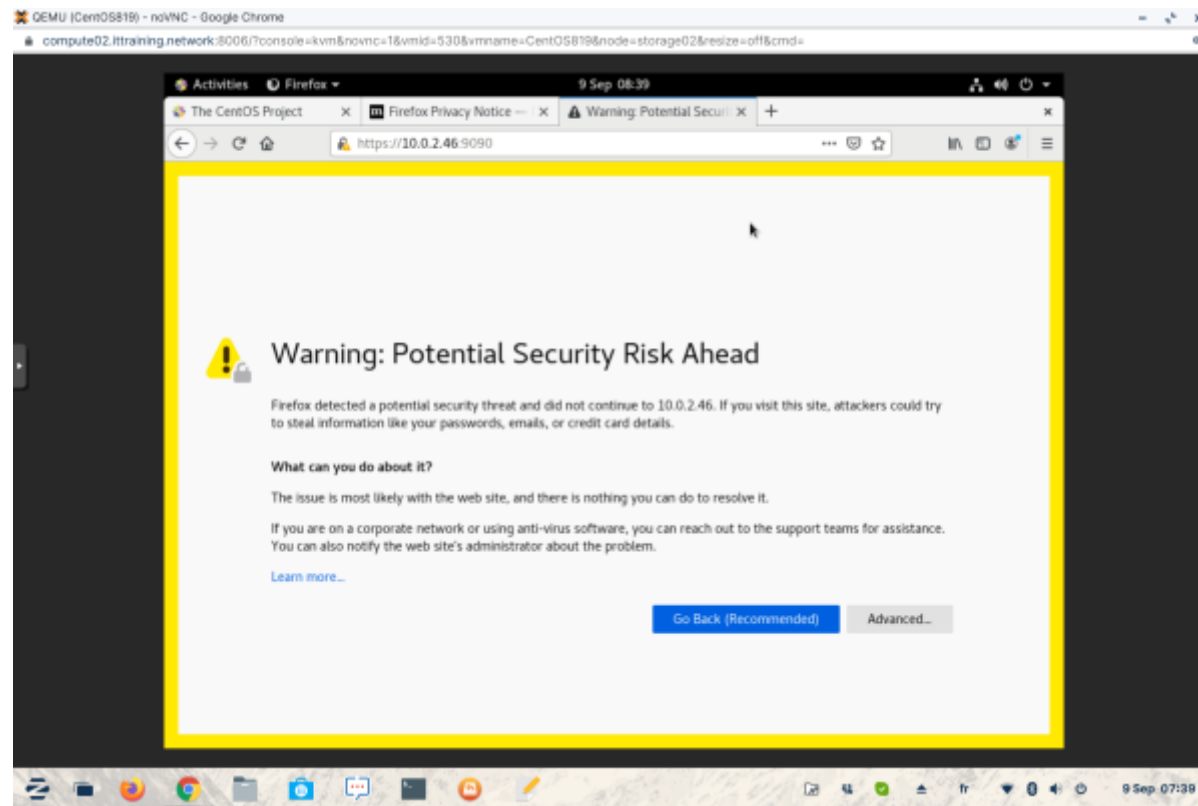
Sep 09 02:24:42 centos8.ittraining.loc systemd[1]: Starting Cockpit Web Service Socke>
Sep 09 02:24:42 centos8.ittraining.loc systemd[1]: Listening on Cockpit Web Service S>
[q]
```

Le cas échéant, ajoutez une règle dans le pare-feu afin de pouvoir utiliser **cockpit** :

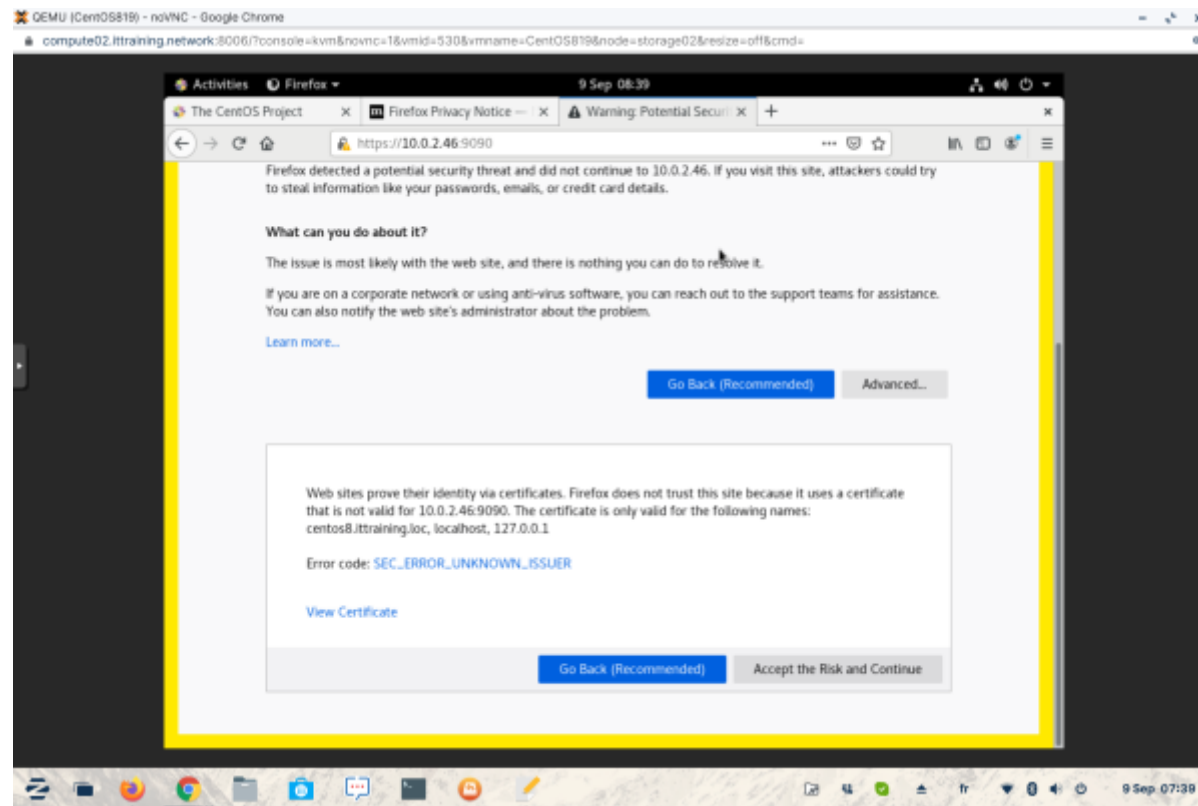
```
[root@centos8 ~]# firewall-cmd --add-service=cockpit --permanent
Warning: ALREADY_ENABLED: cockpit
success
```

6.2 - Connexion à l'Interface

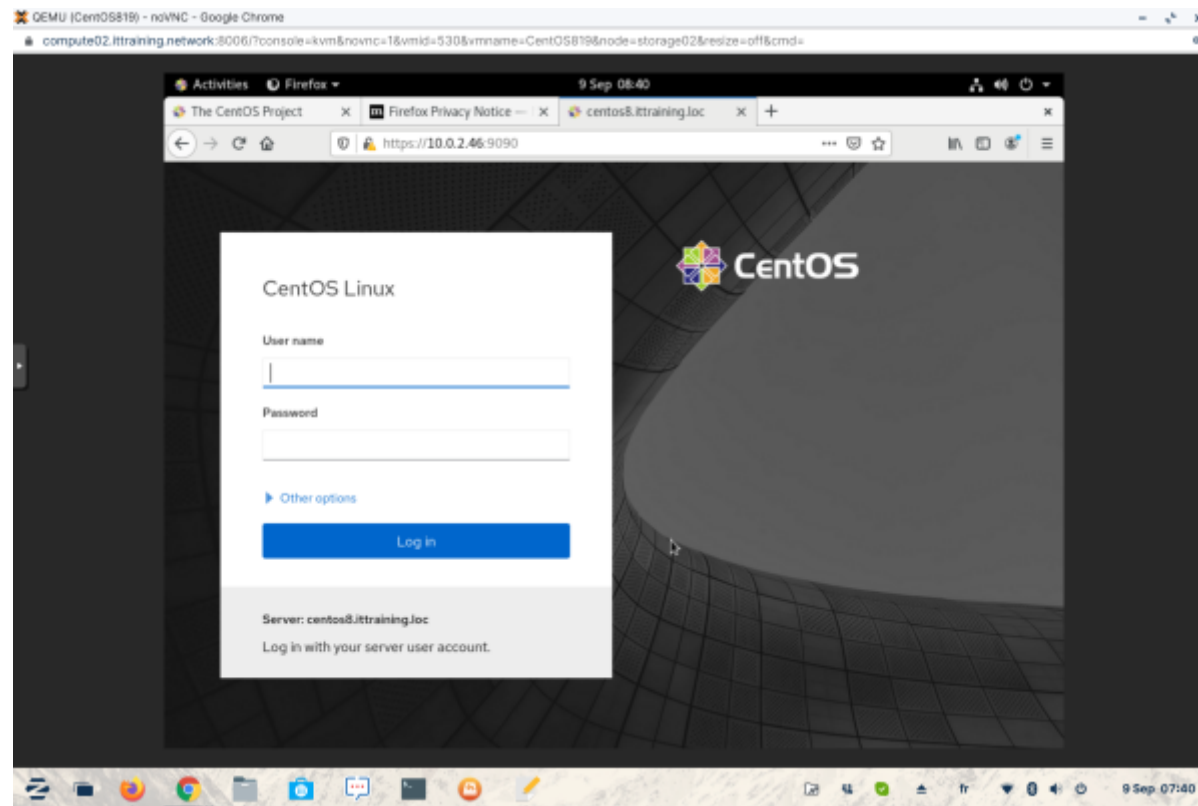
Ouvrez la connexion **CentOS8_VNC_10.0.2.46** dans l'interface de Guacamole. Connectez-vous en tant que **trainee** et ouvrez le navigateur web. Saisissez l'adresse <https://10.0.2.46:9090> :



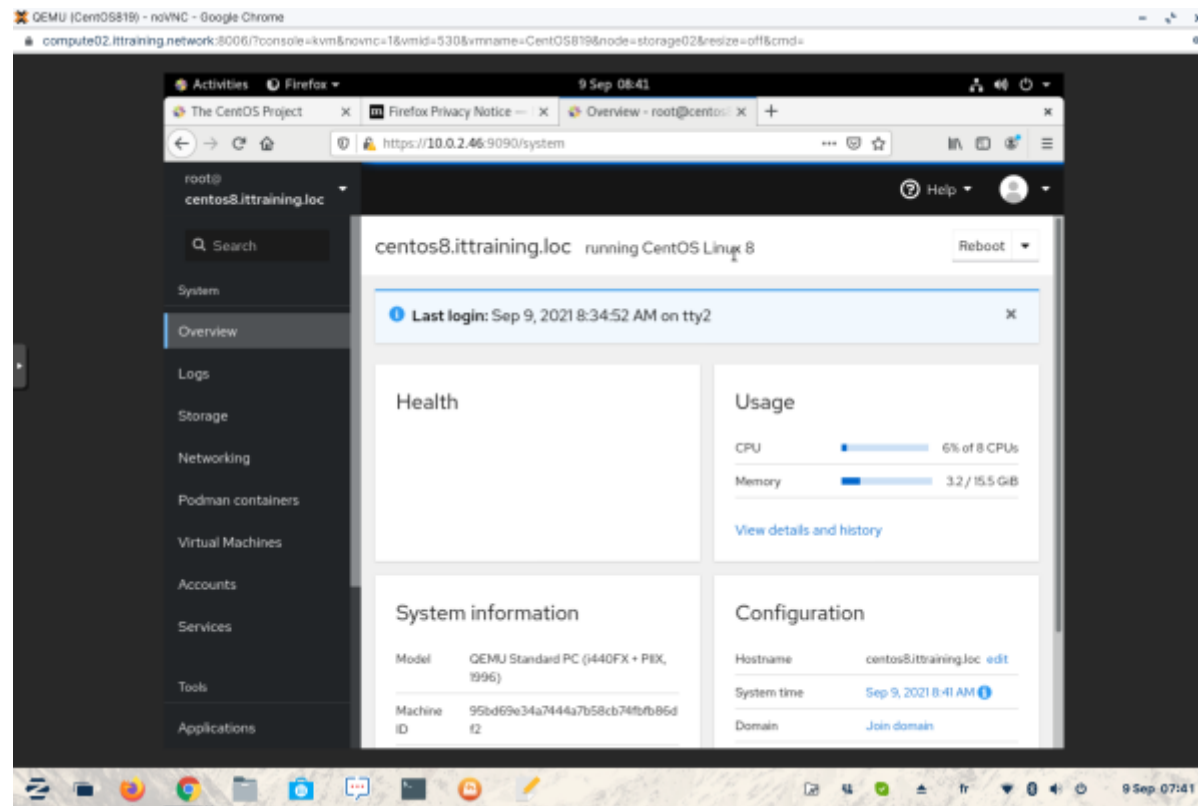
Cliquez sur le bouton **Advanced** :



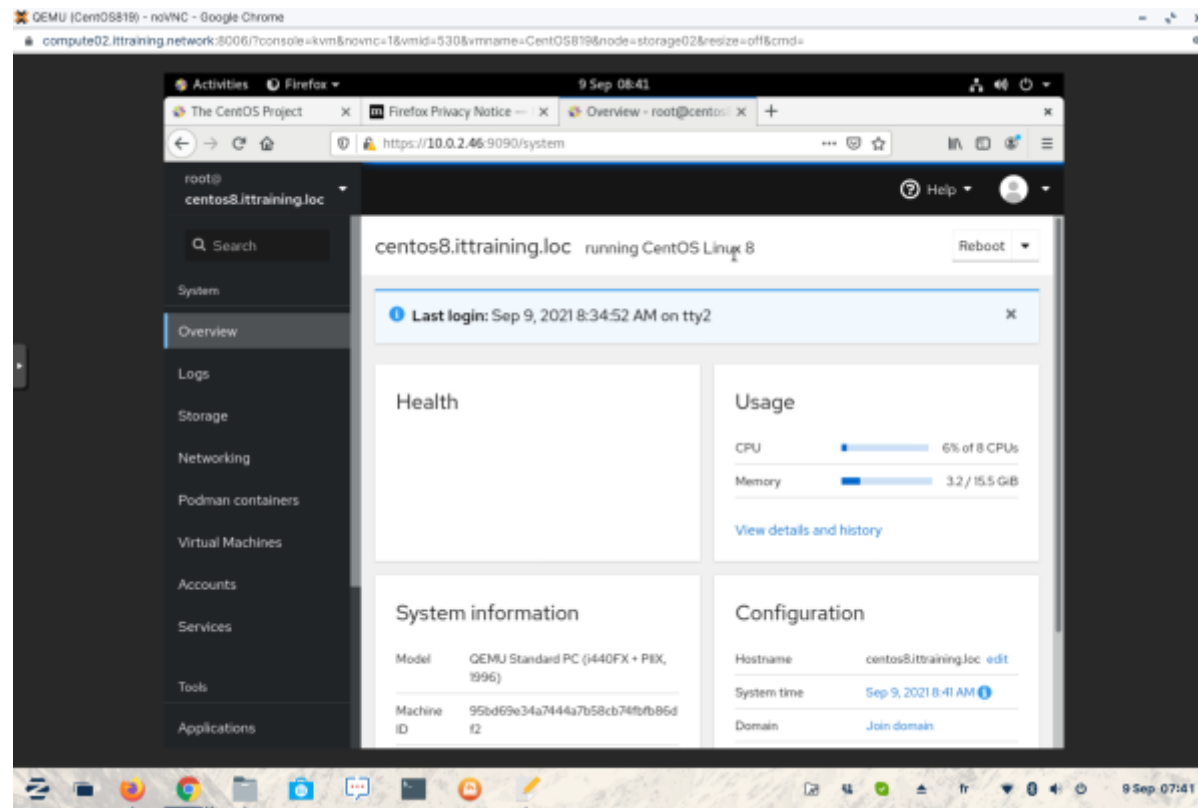
Cliquez sur le bouton **Accept the Risk and Continue** :



Entrez les coordonnées de connexion **root** / **fenestros** et cliquez sur le bouton **Log in** :

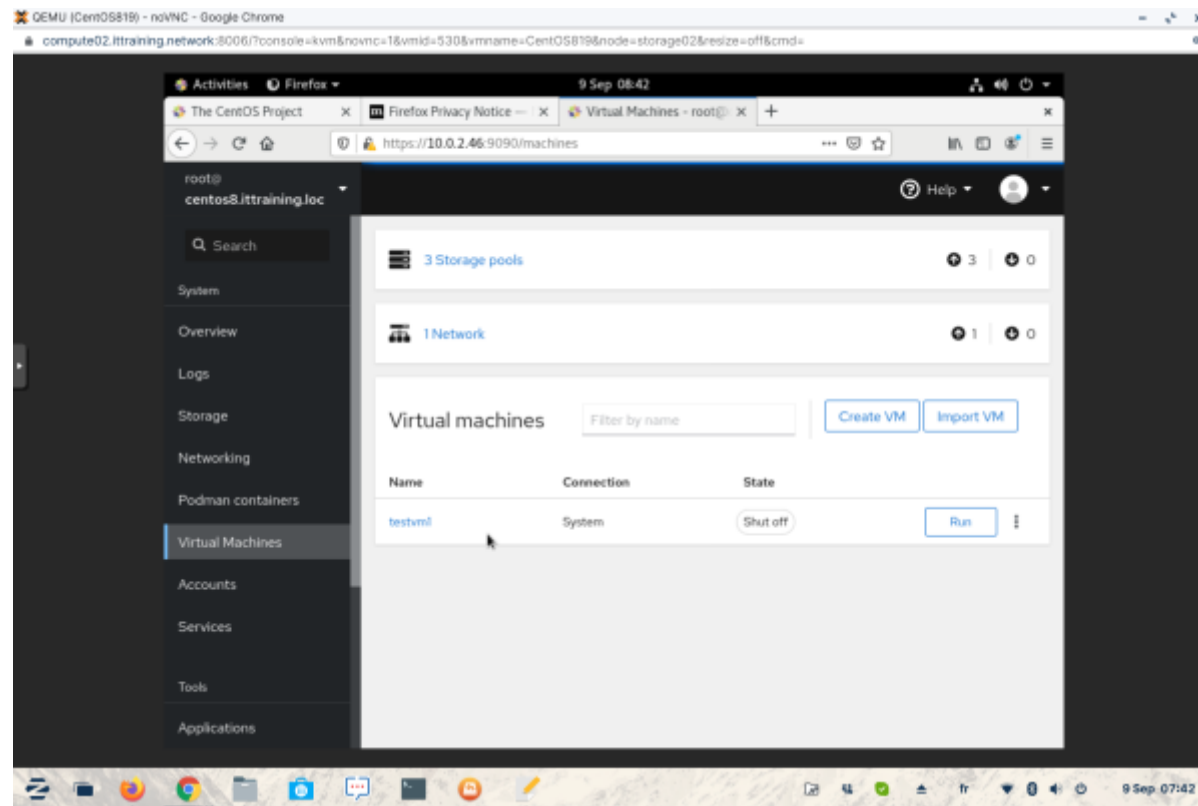


La section **Overview** vous donne des informations sur le système d'exploitation de votre hôte KVM :

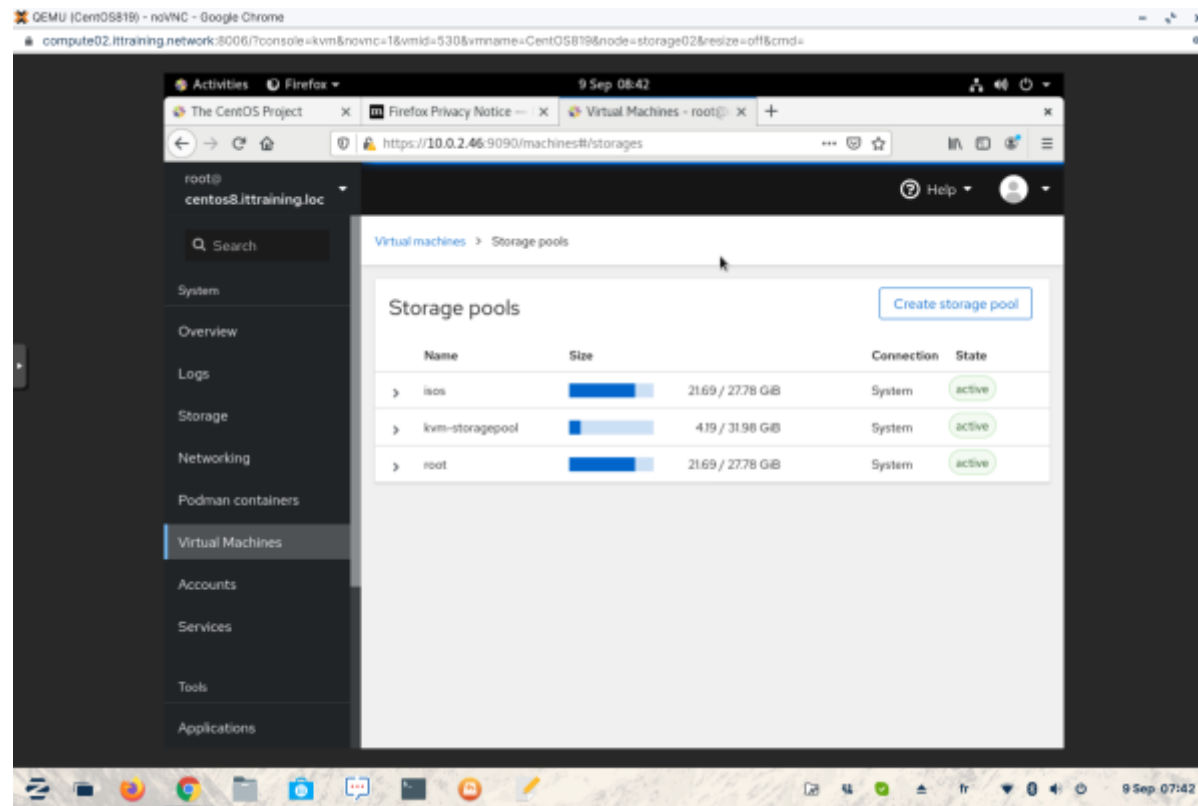


6.3 - Visualisation de la Configuration de la VM testvm1

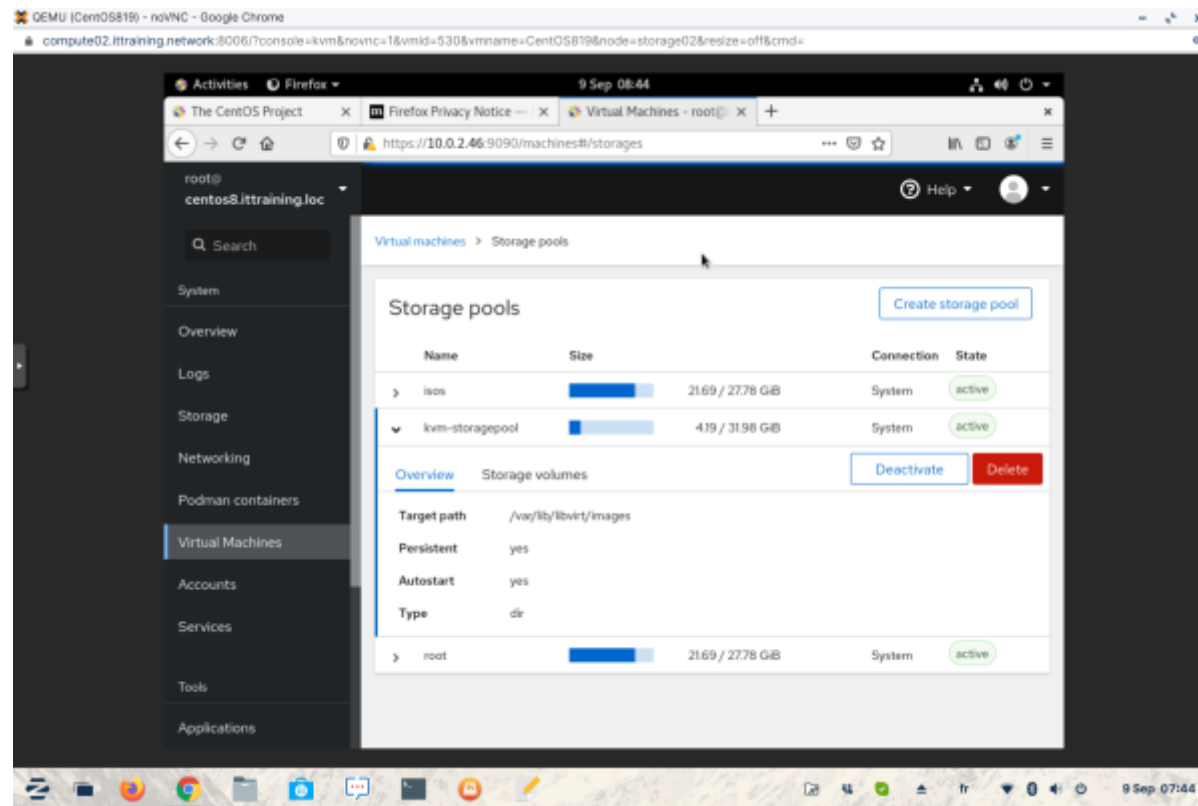
Cliquez ensuite sur **Virtual Machines** dans la colonne de gauche :



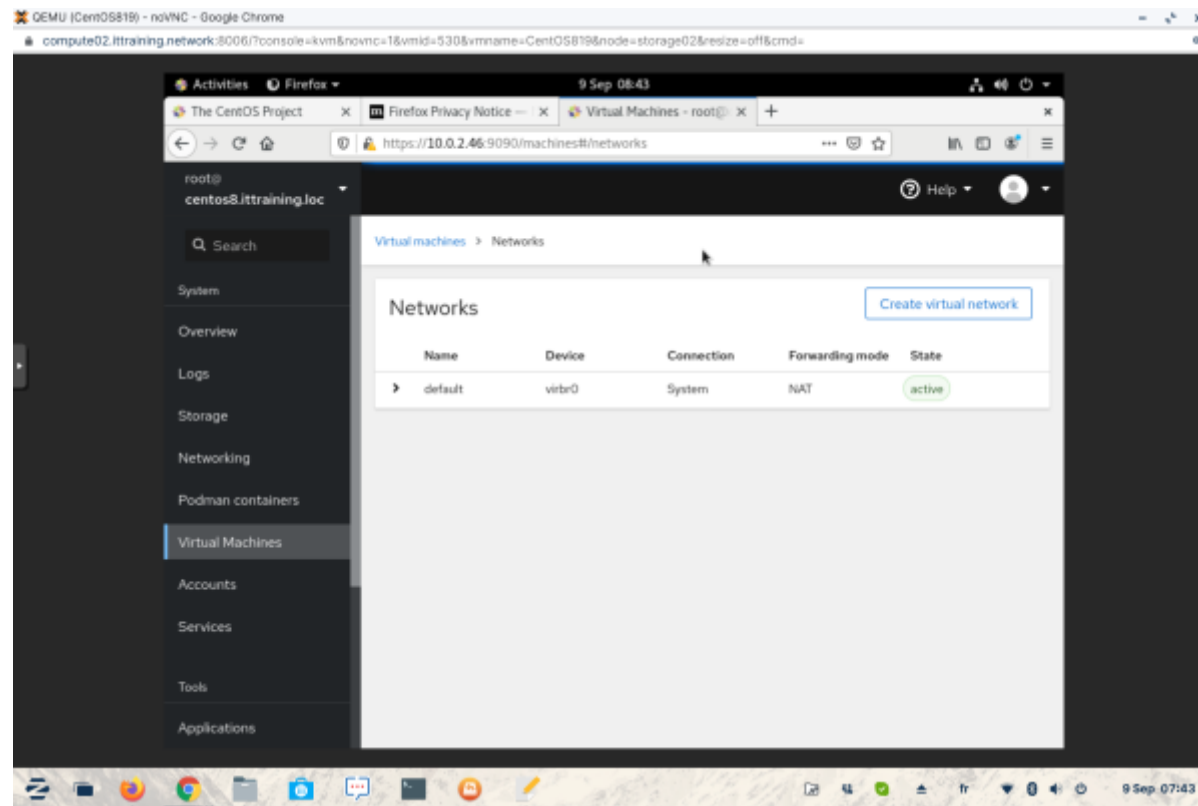
Cliquez maintenant sur le lien **3 Storage pools** :



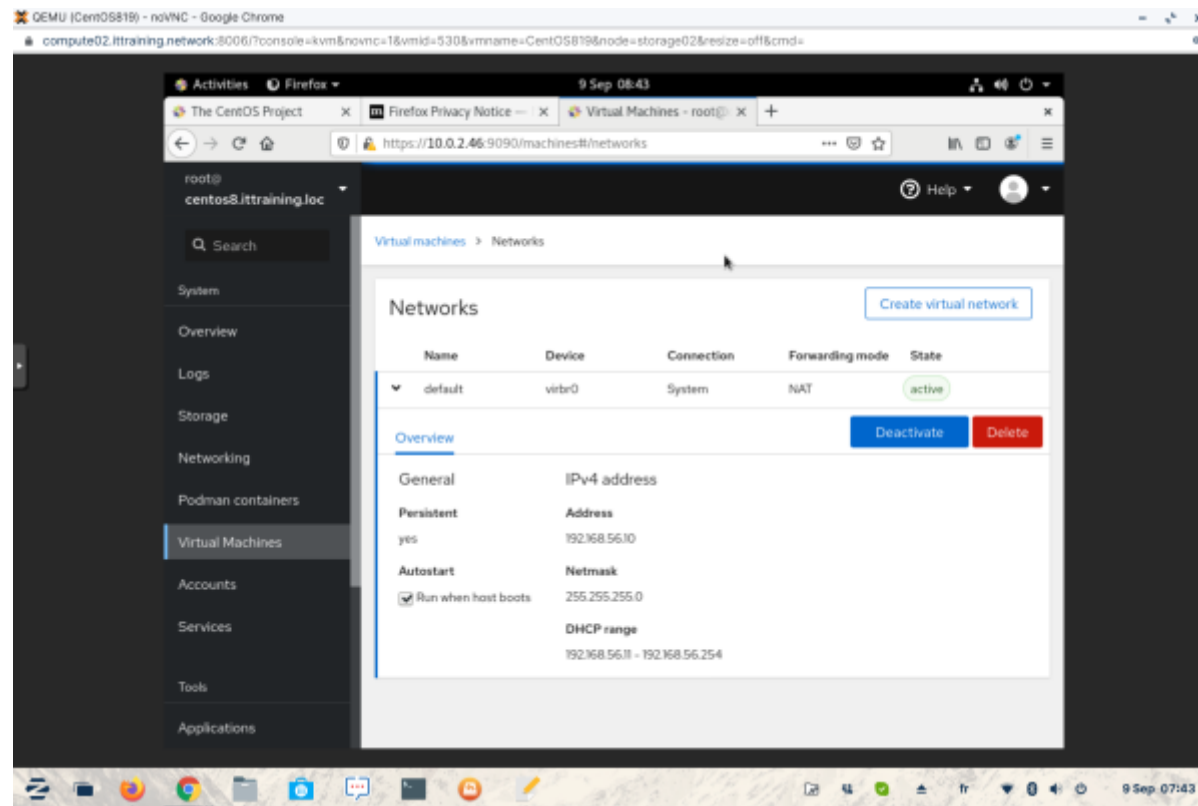
Pour consulter les informations concernant le storage pool **kvm-storagepool**, cliquez sur celui-ci :



Cliquez ensuite sur les liens **Virtual machines > Networks** :

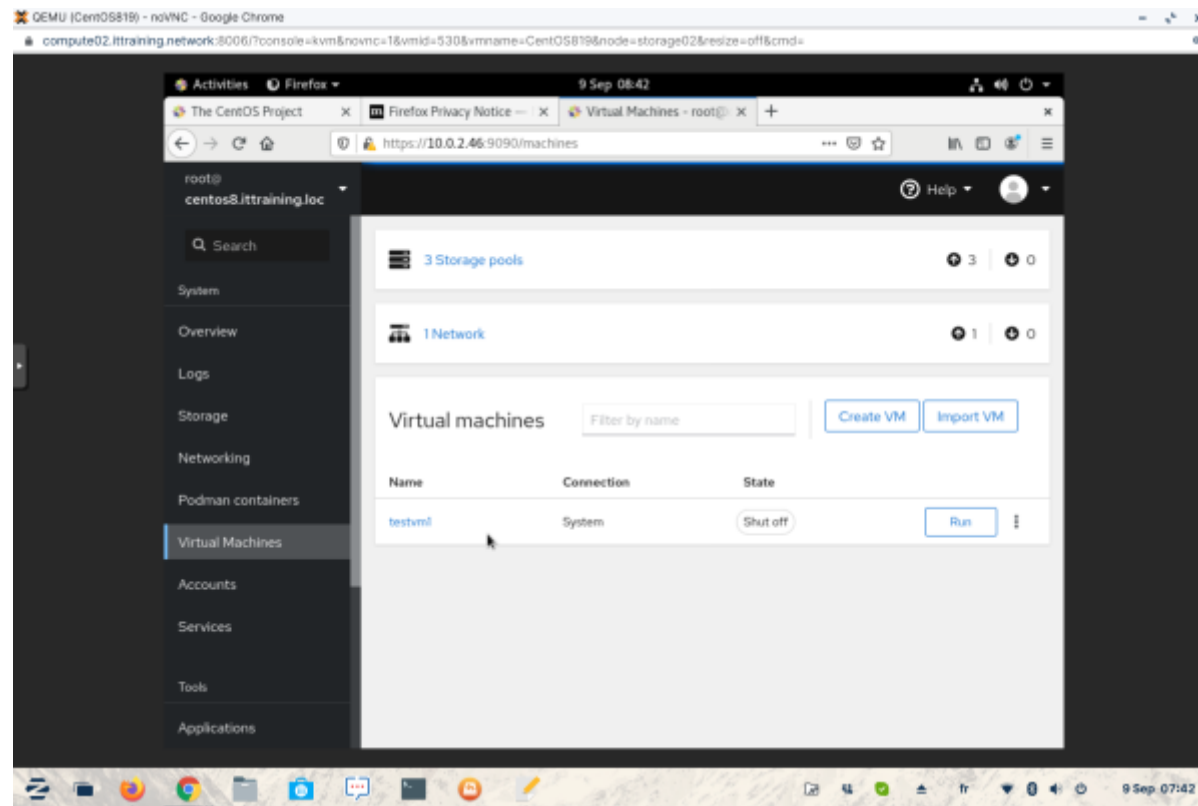


En cliquant sur le lien **default** vous obtiendrez des informations concernant le pont configuré :

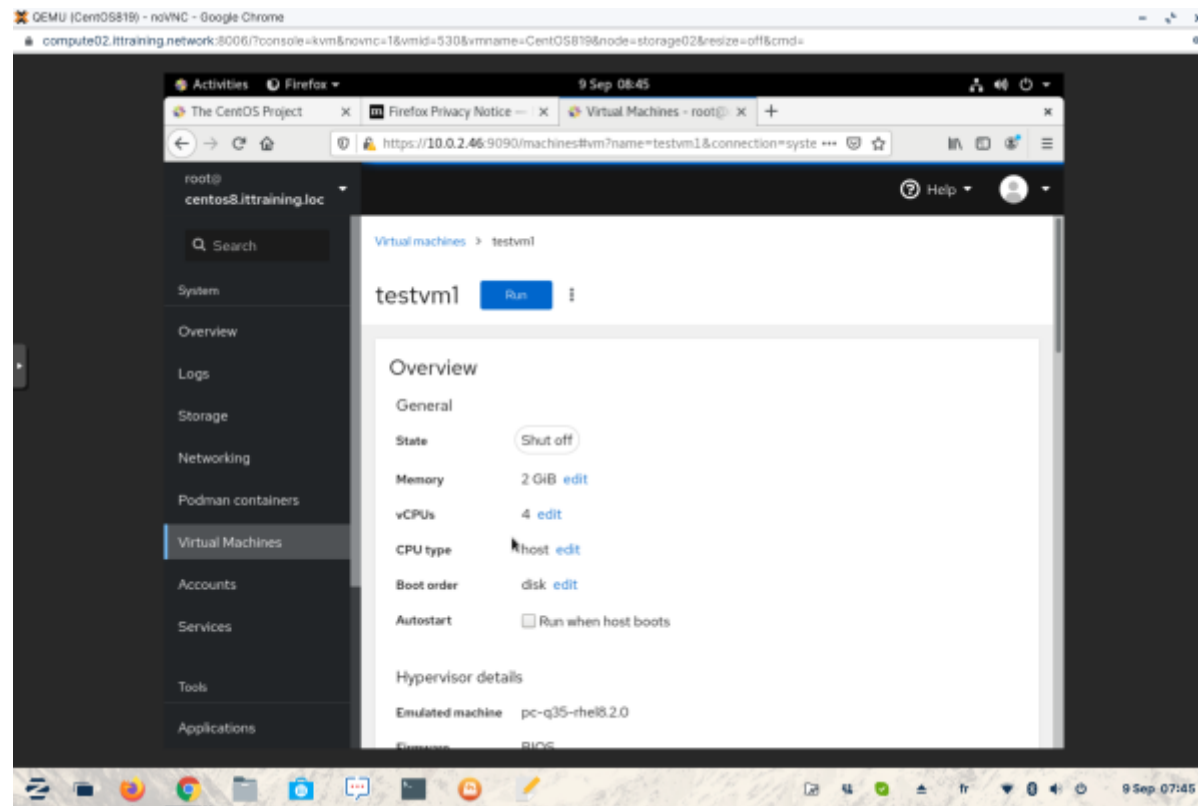


6.4 - Démarrage de la VM testvm1

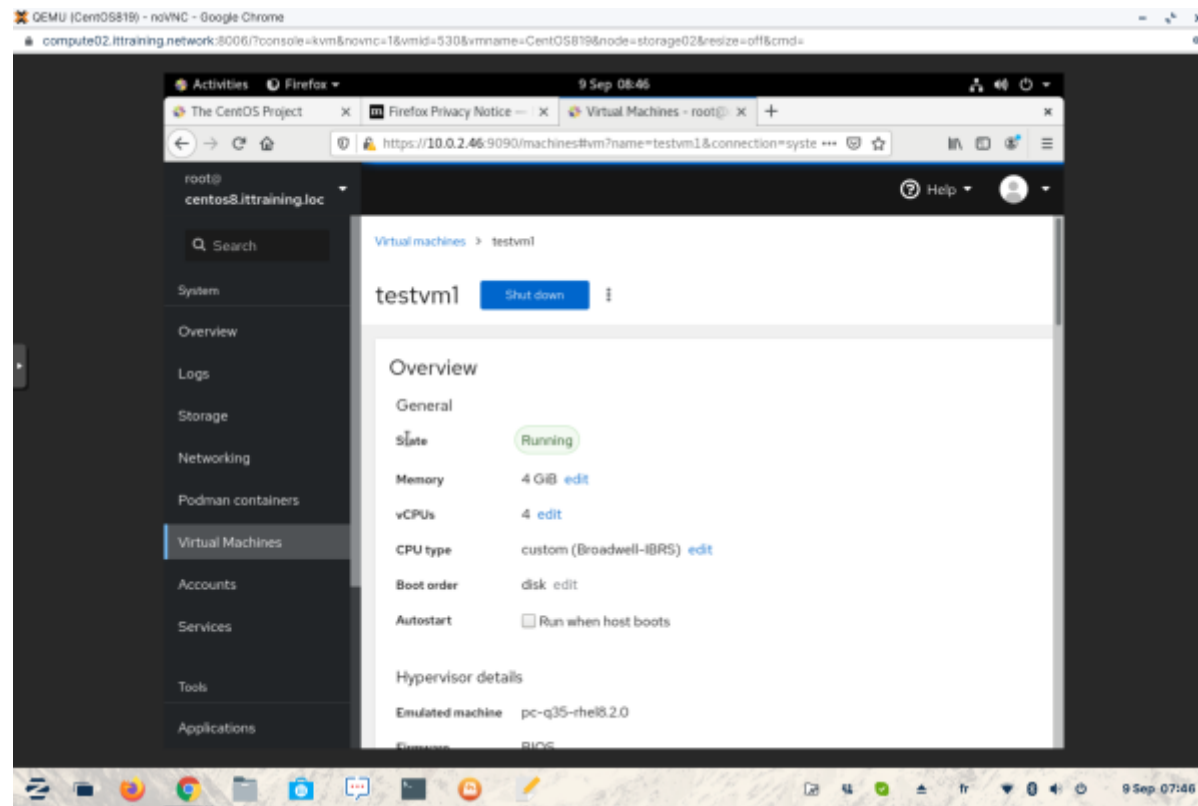
Cliquez ensuite sur le lien **Virtual machines** :



Cliquez sur le lien **testvm1** :

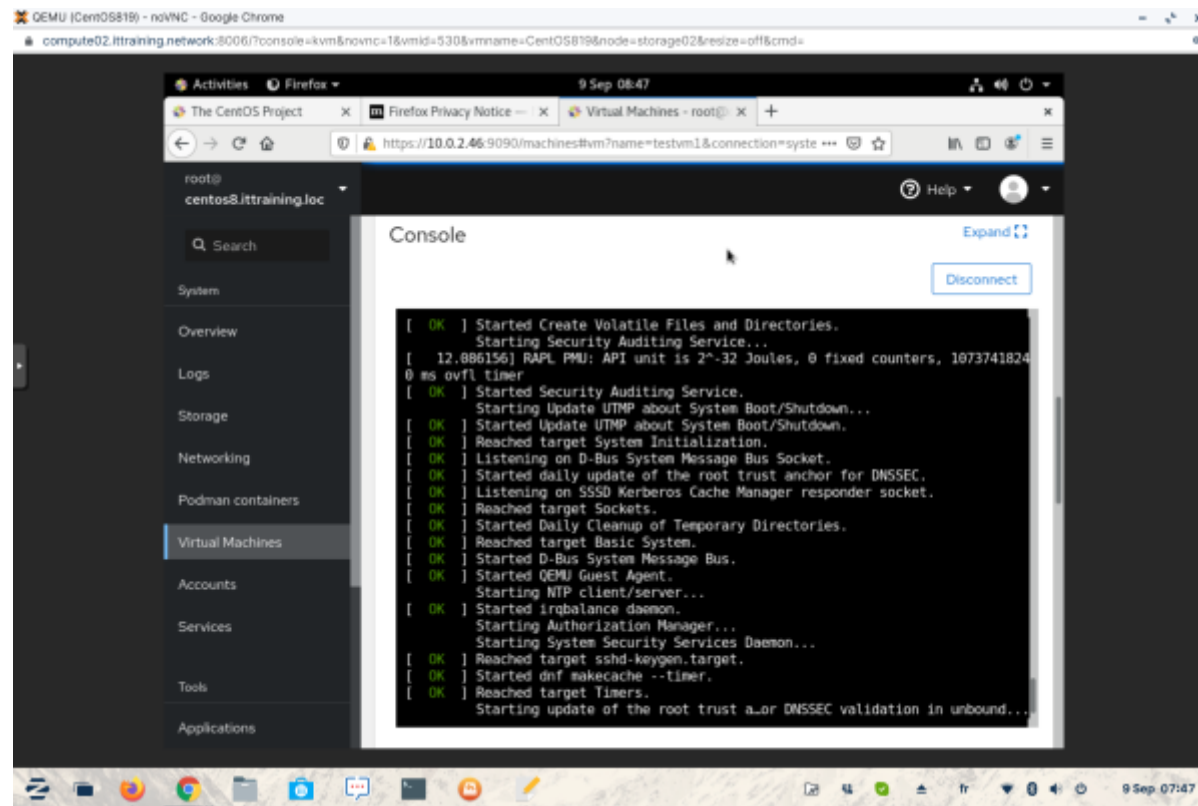


Cliquez sur le bouton **Run** pour démarrer la VM :

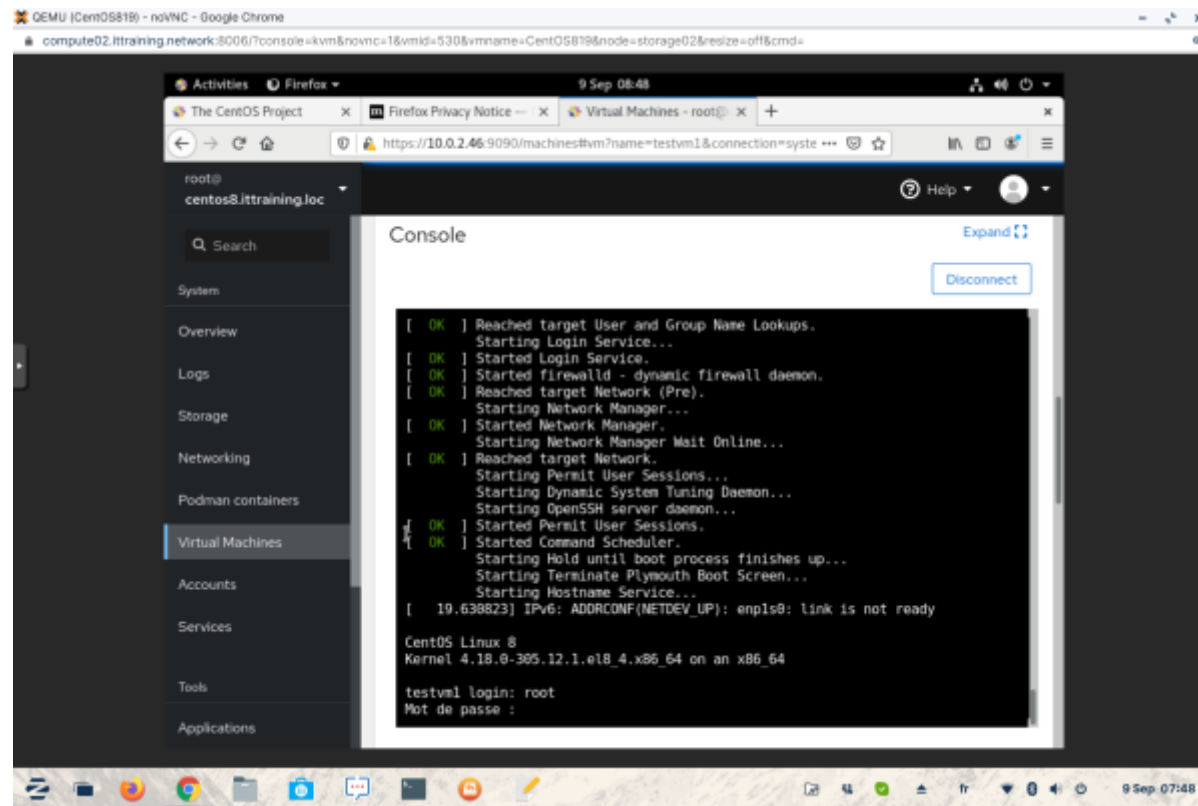


6.5 - Connexion à la VM testvm1

Descendez la fenêtre et cliquez sur le lien **Console** :



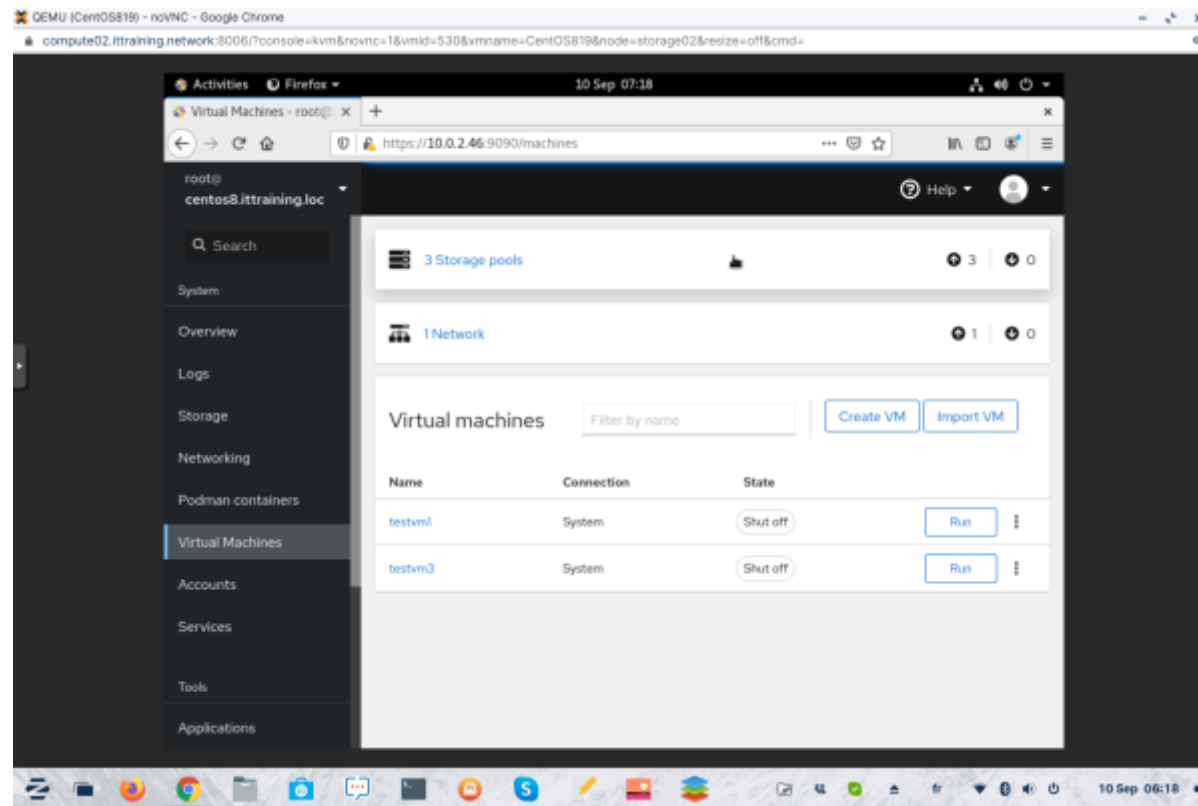
A l'issu du démarrage, connectez-vous à la VM :



LAB #7 - Gestion des Machines Virtuelles avec Cockpit

7.1 - Création d'un Clone de la VM testvm1

A l'aide de l'interface cockpit, créez puis supprimez un clone **testvm3** de la machine virtuelle **testvm1** :



7.2 - Création d'une VM en Mode Graphique à partir d'un ISO

A l'aide de l'interface cockpit, installez une nouvelle machine virtuelle, **testvm2** ayant 2vCPUs et 2Go de RAM, à partir de l'ISO et procédez à sa configuration.

7.3 - Création et Suppression d'un Snapshot de la vm testvm2

A l'aide de l'interface cockpit, créez puis supprimez un snapshot de la machine virtuelle, **testvm2**,

7.4 - Augmentation des Ressources de la VM testvm2

A l'aide de l'interface cockpit, augmentez le nombre de vCPUs de la machine virtuelle testvm2 à **4** puis augmentez la mémoire allouée à la machine virtuelle à 4 Go.

7.5 - Ajout et Suppression d'un Disque Dur à la VM testvm2

A l'aide de l'interface cockpit, ajoutez un disque dur de 20 Go à la machine virtuelle testvm2.



A Faire : Supprimez la machine virtuelle testvm2. Assurez-vous que votre machine virtuelle **testvm1** soit arrêtée.

LAB #8 - Gestion de KVM avec virt-manager

8.1 - Installation

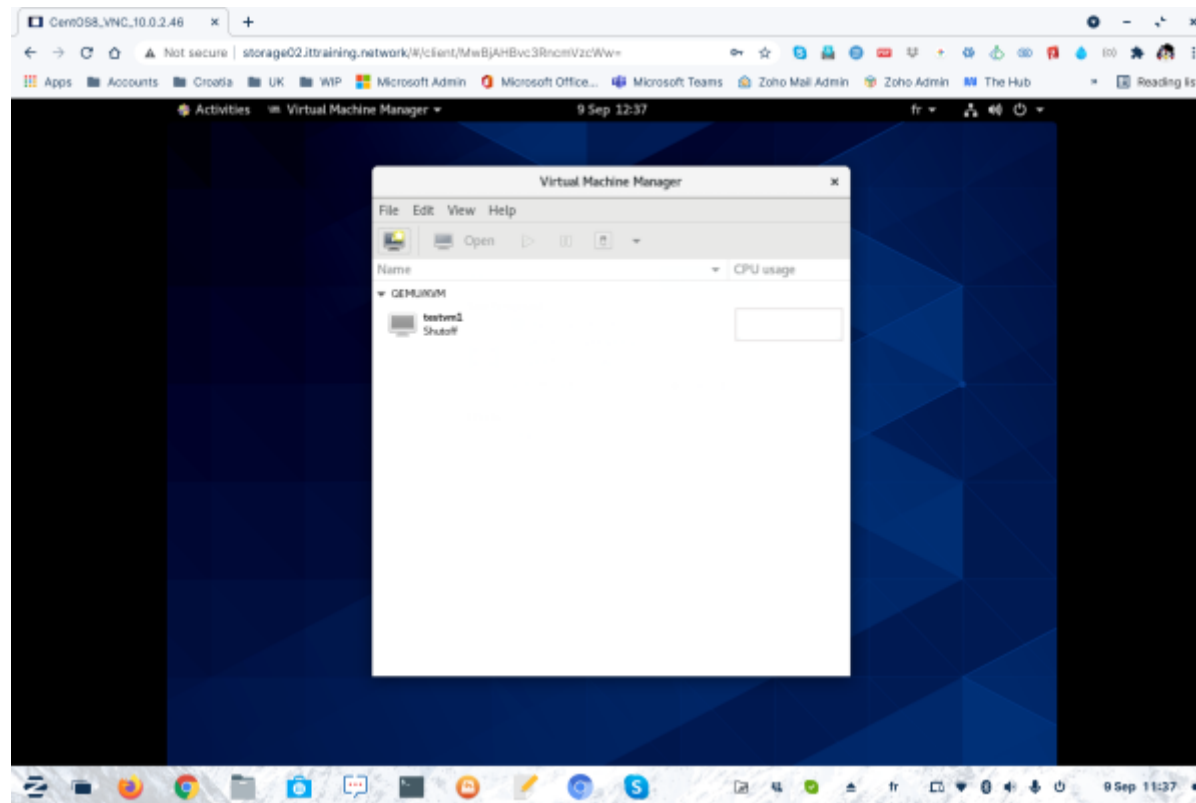
L'exécutable **virt-manager** n'est pas installée par défaut. Cherchez donc le paquet pouvant servir à l'installation de l'exécutable et installez-la :

```
[root@centos8 ~]# dnf provides virt-manager
Last metadata expiration check: 1:14:32 ago on Thu 09 Sep 2021 11:19:54 CEST.
virt-manager-2.2.1-4.el8.noarch : Desktop tool for managing virtual machines via
                                : libvirt
Repo                            : appstream
Matched from:
Provide                         : virt-manager = 2.2.1-4.el8
```

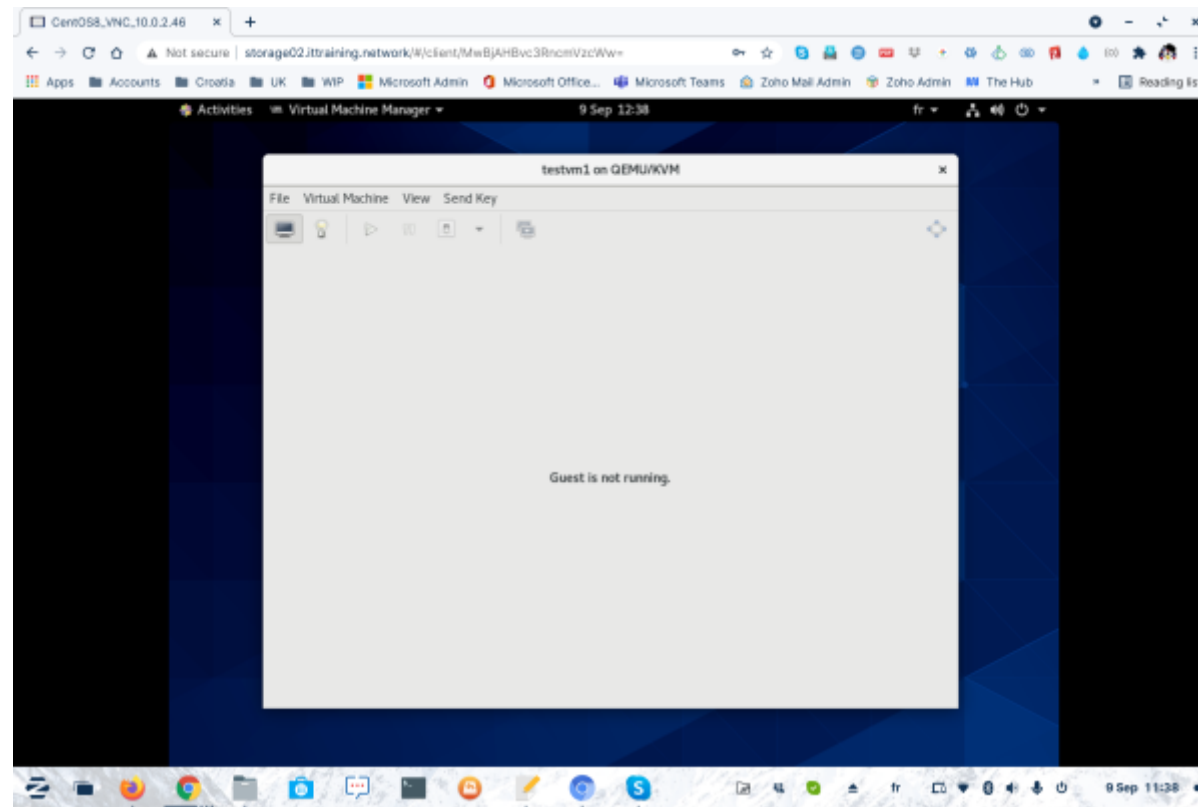
```
[root@centos8 ~]# dnf install virt-manager
```

8.2 - Connexion à l'Interface

Exécutez ensuite **virt-manager**. Vous verrez une liste des machines virtuelles :

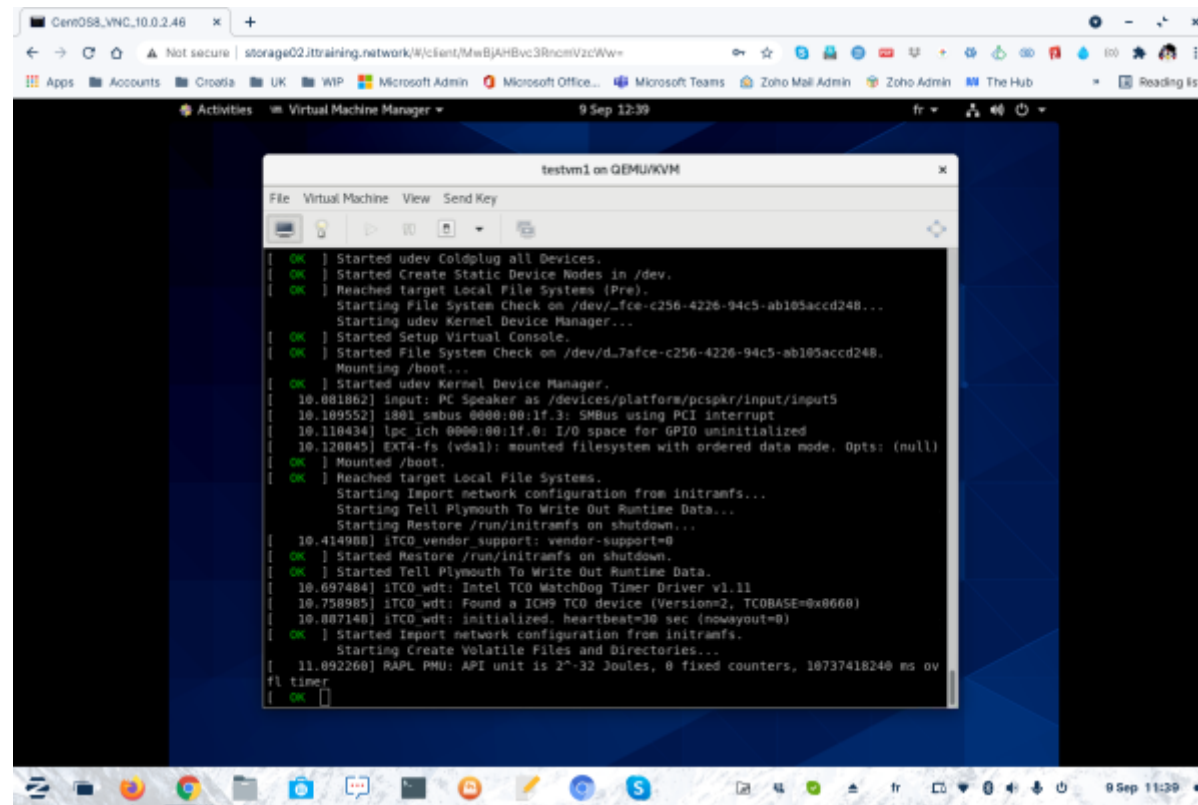


Cliquez sur **testvm1** :



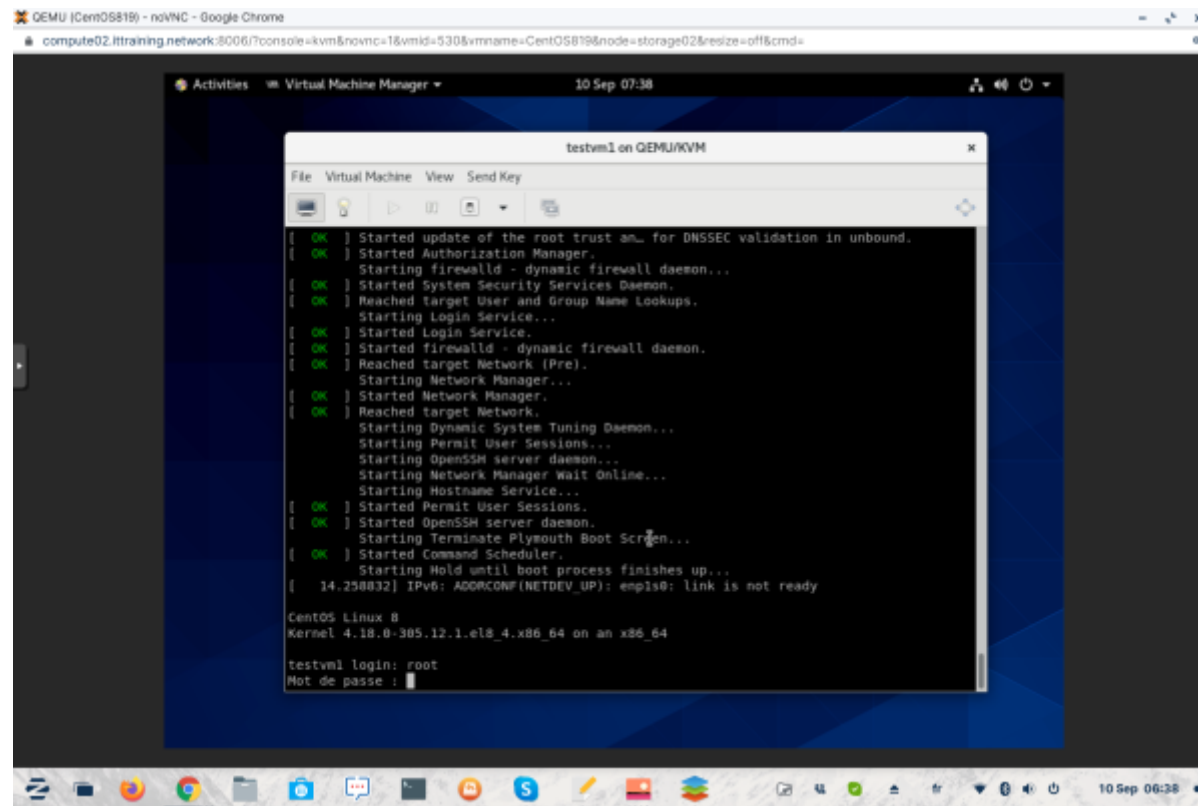
8.3 - Démarrage de la VM testvm1

Cliquez sur l'icône pour démarrer la machine virtuelle :



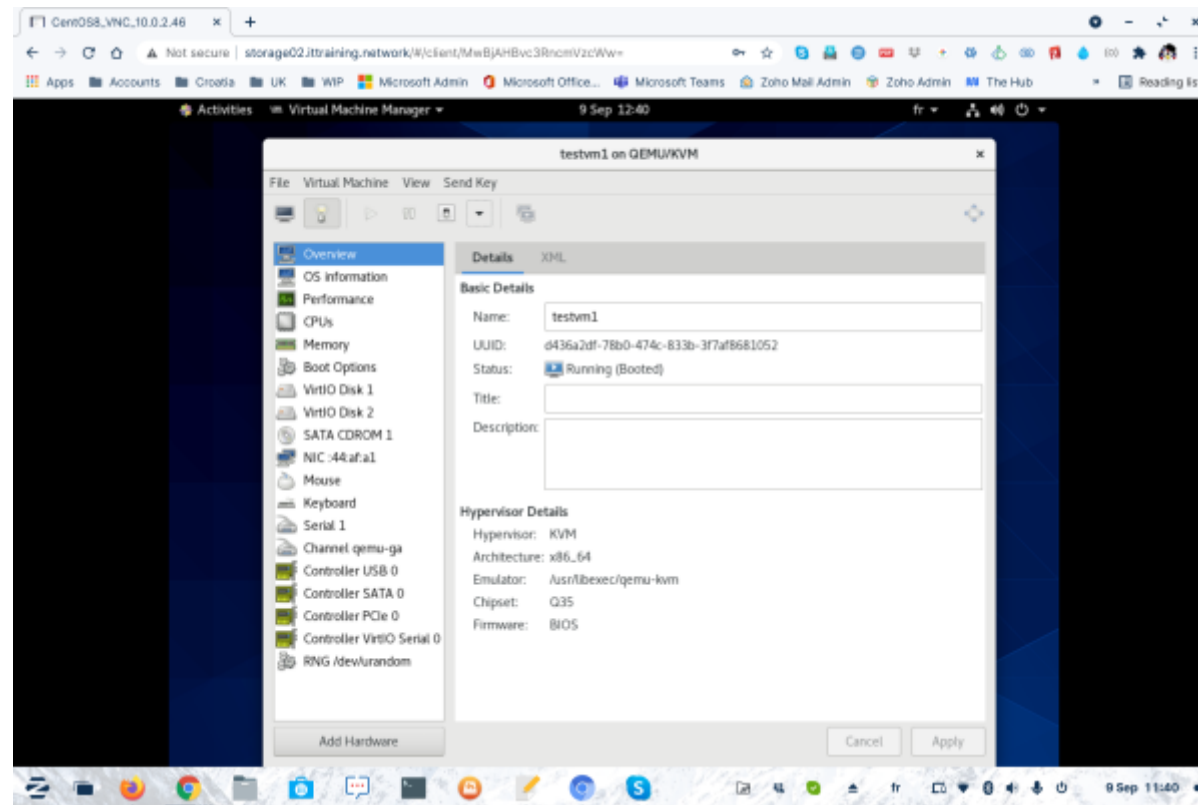
8.4 - Connexion à la VM testvm1

A l'issu du démarrage, connectez-vous à la machine virtuelle :

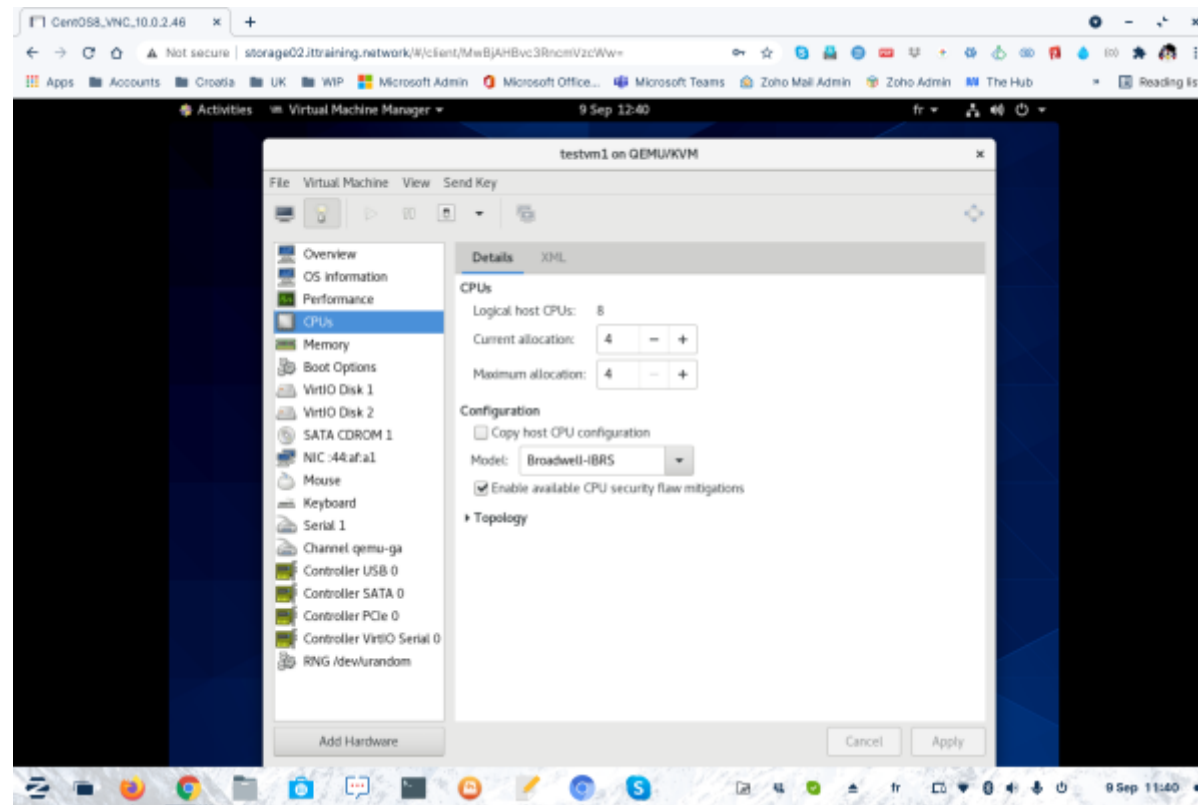


8.5 - Visualisation de la Configuration de la VM

Cliquez sur l'icône de l'ampoule pour visualiser la configuration de la machine virtuelle :



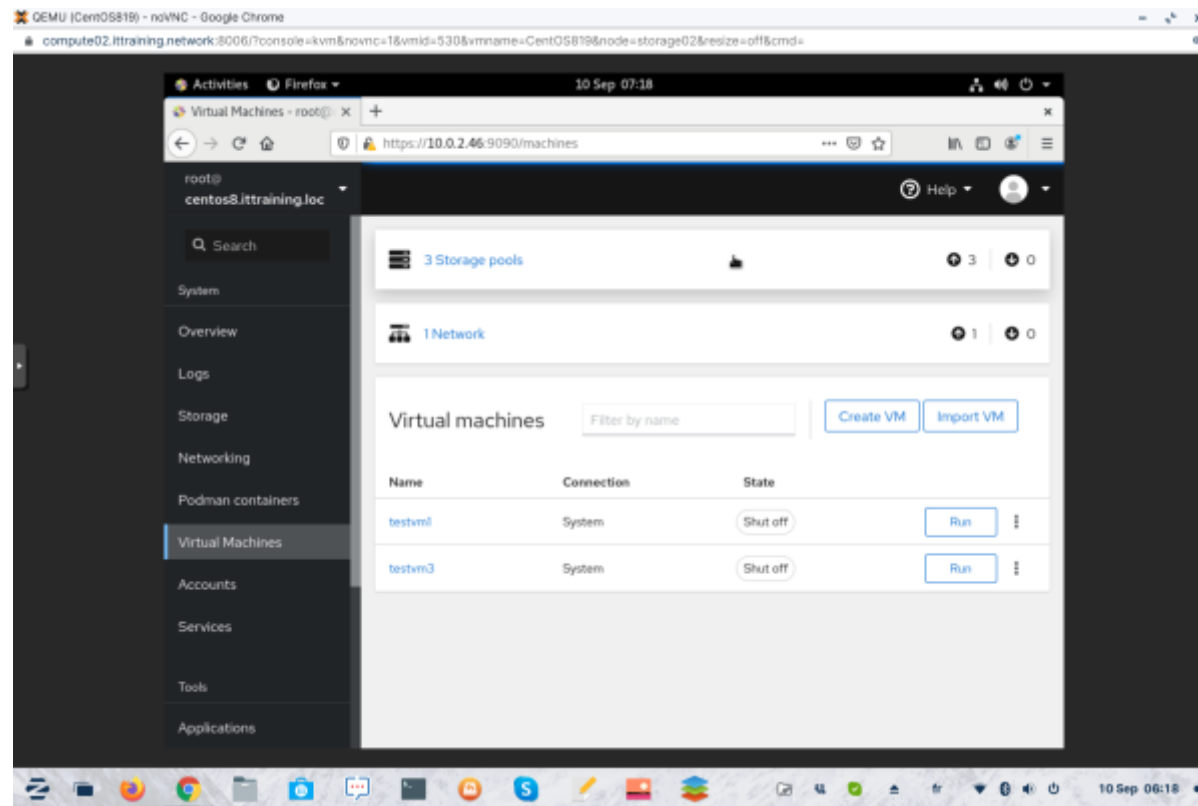
Cliquez ensuite sur **CPUs** dans la colonne de gauche pour visualiser les détails des vCPUs de la machine virtuelle :



LAB #9 - Gestion des Machines Virtuelles avec virt-manager

9.1 - Création d'un Clone de la VM testvm1

A l'aide de l'interface virt-manager, créez puis supprimez un clone **testvm3** de la machine virtuelle **testvm1** :



9.2 - Création d'une VM en Mode Graphique à partir d'un ISO

A l'aide de l'interface virt-manager, installez une nouvelle machine virtuelle, **testvm2** ayant 2vCPUs et 2Go de RAM, à partir de l'ISO et procédez à sa configuration.

9.3 - Création et Suppression d'un Snapshot de la vm testvm2

A l'aide de l'interface virt-manager, créez puis supprimez un snapshot de la machine virtuelle, **testvm2**,

9.4 - Augmentation des Ressources de la VM testvm2

A l'aide de l'interface virt-manager, augmentez le nombre de vCPUs de la machine virtuelle testvm2 à **4** puis augmentez la mémoire allouée à la machine virtuelle à 4 Go.

9.5 - Ajout et Suppression d'un Disque Dur à la VM testvm2

A l'aide de l'interface virt-manager, ajoutez un disque dur de 20 Go à la machine virtuelle testvm2.



A Faire : Supprimez la machine virtuelle testvm2. Assurez-vous que votre machine virtuelle **testvm1** soit arrêtée.

<html>

Copyright © 2022 Hugh Norris

</html>
