

Configurer la machine virtuelle Windows au module CGM-SRV sur CGR1xxx

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Informations générales](#)

[Configuration](#)

[Créer une image de machine virtuelle Windows](#)

[Installer KVM sur votre machine Linux](#)

[Vérifier l'installation de KVM](#)

[Créer une machine virtuelle Windows](#)

[Déployer une image de machine virtuelle Windows sur CGM-SRV](#)

[Vérification](#)

[Dépannage](#)

Introduction

Ce document décrit les étapes nécessaires pour créer et exécuter une machine virtuelle Windows (VM) sur le module CGM (Connected Grid Module) - Serveur système (SRV).

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Linux
- Machine virtuelle basée sur le noyau (KVM)
- Comprendre les concepts de virtualisation

Components Used

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Routeurs de la grille connectée (CGR) 1120
- Module CGM-SRV-XX
- Les étapes de configuration de CGM-SRV sont exécutées avant ce guide :
- ISO d'installation de Windows 7
- Visionneuse VNC (Virtual Network Computing)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Informations générales

Lorsque vous souhaitez exécuter des applications IOx ou des machines virtuelles sur la plate-forme CGR1000, vous pouvez utiliser le module de calcul CGM-SRV. Le module CGM-SRV est en fait un petit serveur qui contient un processeur x86 multicoeur, de la mémoire et du stockage. Les modèles CGR1120 et CGR1240 peuvent tous deux être équipés de l'un de ces modules pour ajouter des fonctionnalités IOx.

Au moment de la rédaction du présent rapport, il existe deux types de documents :

| Unité de conservation des stocks (SKU) | Disques SSD (Solid State Drives) | RAM UC |
|--|----------------------------------|-----------------------|
| CGM-SRV-64 | 64 Go (50 Go utilisables) | 4 Go 4 coeurs 800 MHz |
| CGM-SRV-128 | 128 Go (100 Go utilisables) | 4 Go 4 coeurs 800 MHz |

Chaque module dispose également de deux ports USB pour le stockage et de sa propre interface Gigabit Ethernet externe.

Comme pour tout autre périphérique compatible IOx, le module peut héberger différents types d'applications IOx, mais en raison de la plus grande capacité du module CGM-SRV, il peut également exécuter une distribution Linux standard ou Windows entièrement configurée (par exemple Ubuntu ou CentOS).

Configuration

Créer une image de machine virtuelle Windows

Pour déployer une machine virtuelle Windows sur le module CGM-SRV, vous devez d'abord créer une image au format QEMU QCOW qui contient l'installation de Windows. Une façon de créer une telle image est d'utiliser KVM et virsh sur une machine Linux.

Les étapes mentionnées ci-dessus n'impliquent pas du tout le CGR1xxx ou CGM-SRV, elles sont simplement requises pour créer une image QCOW de machine virtuelle Windows 7 de base que vous pouvez déployer à l'étape suivante de CGM-SRV.

Pour ce guide, vous pouvez commencer avec une installation CentOS7 minimale fraîchement installée. Les étapes pour les autres distributions Linux doivent être similaires mais peuvent légèrement différer.

Installer KVM sur votre machine Linux

Étape 1. La première chose à faire est de vérifier si la machine hôte prend en charge les extensions VM. Sur la plate-forme x86, il s'agit d'AMD-V ou de VT-X d'Intel. La plupart, sinon la totalité, des processeurs x86 modernes prennent en charge ces extensions. Même lorsque vous exécutez une machine virtuelle, la plupart des hyperviseurs offrent la possibilité de transmettre/émuler ces extensions.

Afin de vérifier si le processeur installé prend en charge ces extensions, vous devez vérifier si l'indicateur vmx (pour VT-X) ou svm (pour AMD-V) existe dans la sortie cpuinfo.

```
[root@cen7 ~]# egrep -c '(vmx|svm)' /proc/cpuinfo
2
```

Si la sortie de cette commande est 0, cela signifie qu'aucun processeur trouvé ne prend en charge les extensions de VM. Dans ce cas, vous pouvez vérifier si ces extensions sont activées dans votre BIOS ou hyperviseur lorsque vous utilisez une machine virtuelle pour exécuter cette machine.

Étape 2. L'étape suivante consiste à créer un pont pour fournir un réseau à la machine virtuelle que vous pouvez exécuter sur KVM.

Premièrement, vous devez activer le transfert IP dans le noyau :

```
[root@cen7 ~]# echo "net.ipv4.ip_forward = 1"|sudo tee /etc/sysctl.d/99-ipforward.conf
net.ipv4.ip_forward = 1
[root@cen7 ~]# sysctl -p /etc/sysctl.d/99-ipforward.conf
net.ipv4.ip_forward = 1
```

Pour créer le pont, la configuration IP doit passer de l'interface réelle au pont lui-même, car il s'agit de l'interface propriétaire de l'adresse IP.

Une fois l'installation standard terminée, la configuration réseau est dans `/etc/sysconfig/network-scripts` :

```
[root@cen7 ~]# ls -l /etc/sysconfig/network-scripts/ifcfg-*
/etc/sysconfig/network-scripts/ifcfg-eno16777736
/etc/sysconfig/network-scripts/ifcfg-lo
```

Étape 3. Comme vous pouvez le voir, il existe actuellement une interface (en dehors de l'interface de bouclage), appelée eno167777736. Vous devez déplacer la configuration liée à l'IP vers une interface de pont que vous pouvez appeler virbr0 :

```
[root@cen7 ~]# vi /etc/sysconfig/network-scripts/ifcfg-virbr0
[root@cen7 ~]# cat /etc/sysconfig/network-scripts/ifcfg-virbr0
DEVICE=virbr0
TYPE=BRIDGE
ONBOOT=yes
BOOTPROTO=static
IPADDR=172.16.245.162
NETMASK=255.255.255.0
GATEWAY=172.16.245.2
DNS1=8.8.8.8
```

Étape 4. Après cela, vous devez nettoyer la configuration IP de l'interface réelle et la connecter au pont virbr0 :

```
[root@cen7 ~]# vi /etc/sysconfig/network-scripts/ifcfg-eno16777736
[root@cen7 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eno16777736
UUID=46f0f247-e164-40cc-866b-9133458d9df8
DEVICE=eno16777736
ONBOOT=yes
BRIDGE=virbr0
HWADDR=00:0c:29:ce:96:38
```

Étape 5. Une fois la configuration réseau terminée, vous pouvez continuer et installer KVM :

```
[root@cen7 ~]# sudo yum install kvm virt-manager libvirt virt-install qemu-kvm xauth dejavu-lgc-sans-fonts -y
...
Complete!
```

Étape 6. Une fois l'installation terminée, le mieux est de redémarrer cette machine pour appliquer les nouveaux modules installés et la configuration réseau :

```
[root@cen7 ~]# init 6
```

Vérifier l'installation de KVM

Étape 7. Une fois le redémarrage terminé, vous devez être en mesure d'accéder à la machine sur la (même) adresse IP configurée sur l'interface de pont. Vous devez vérifier si le module du noyau KVM est chargé :

```
root@cen7 ~]# lsmod|grep kvm
kvm_intel                200704  0
kvm                      589824  1 kvm_intel
irqbypass                16384   1 kvm
```

Étape 8. Si cela semble correct, vous pouvez essayer de vous connecter à virsh :

```
[root@cen7 ~]# sudo virsh -c qemu:///system list
 Id      Name                               State
-----
```

Étape 9. Une dernière étape consiste à ouvrir le port 5900 sur le pare-feu de cet ordinateur pour l'accès VNC à l'installation de Windows :

```
[root@cen7 ~]# firewall-cmd --zone=public --add-port=5900/tcp --permanent
success
[root@cen7 ~]# firewall-cmd --reload
success
```

Créer une machine virtuelle Windows

Maintenant que vous disposez d'un système qui fonctionne avec l'installation de KVM, vous pouvez lancer une nouvelle machine virtuelle sur KVM et exécuter les boîtes de dialogue d'installation de Windows.

Étape 1. Copiez l'ISO d'installation de Windows 7 sur votre machine virtuelle (ou rendez-la accessible sur le réseau) :

```
[root@cen7 ~]# scp jedepuyd@172.16.X.X:/home/jedepuyd/win7install.iso /var
jedepuyd@172.16.X.X's password:
win7install.iso                               100% 4546MB  62.1MB/s
01:13
```

Étape 2. Créez une machine virtuelle KVM et laissez-la démarrer à partir de l'ISO de Windows 7 :

```
root@cen7 ~]# virt-install --connect qemu:///system -n win7 -r 1024 --vcpus=2 --disk
```

```
path=/var/lib/libvirt/images/win7.img,size=9 --graphics vnc,listen=0.0.0.0 --noautoconsole --os-  
type windows --os-variant win7 --accelerate --network=bridge:virbr0 --hvm --cdrom  
/var/win7install.iso
```

Starting install...

Allocating win7.img | 9.0 GB

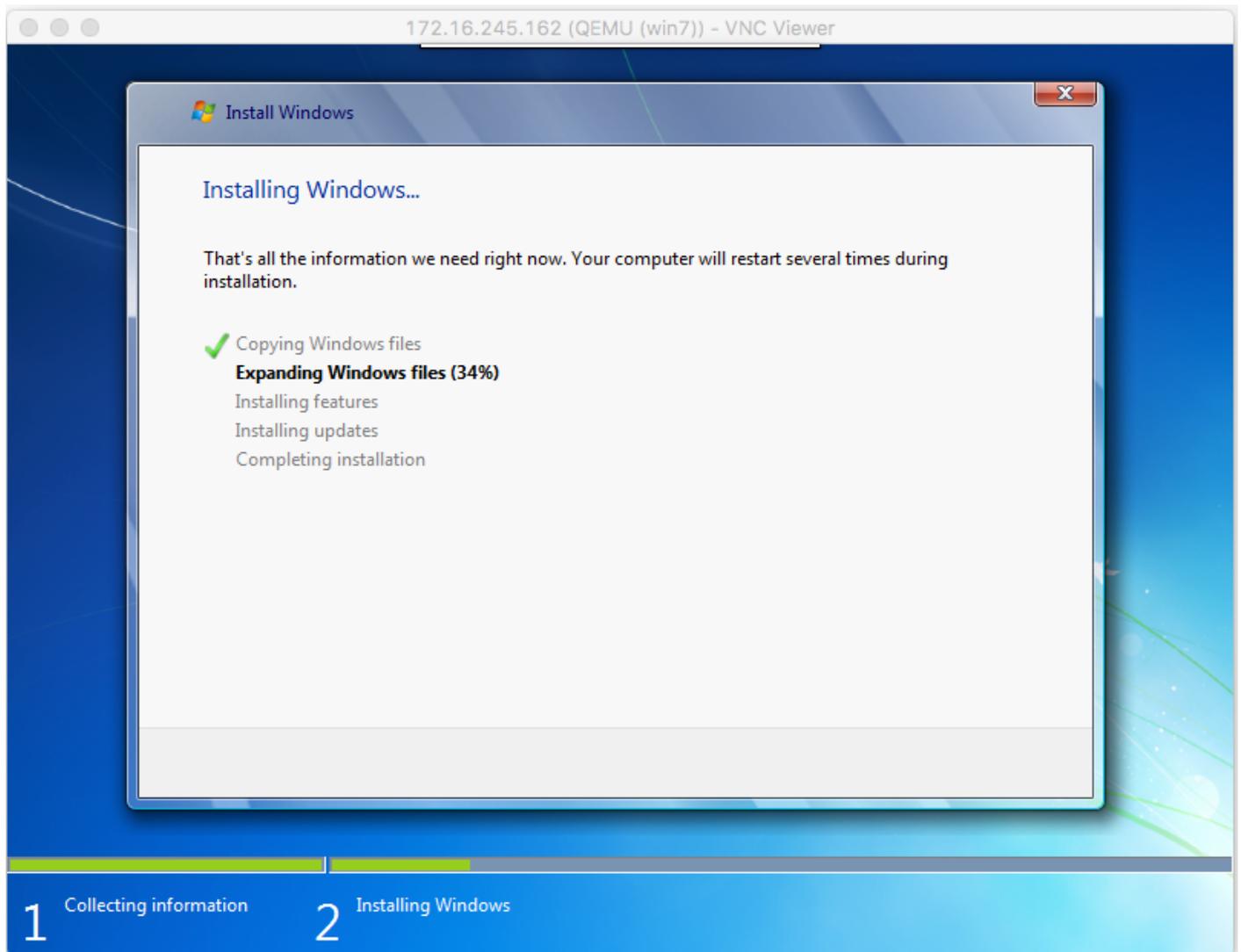
00:00:00

Creating domain... | 0 B

00:00:00

Domain installation still in progress. You can reconnect to
the console to complete the installation process.

Étape 3. Une fois la machine virtuelle démarrée, vous pouvez vous connecter à l'aide de la
visionneuse VNC à l'adresse IP de l'ordinateur hôte sur le port 5900 et terminer l'installation
Windows standard comme illustré sur l'image :



Si Windows redémarre au moment de l'installation, il peut être nécessaire de redémarrer la
machine virtuelle avec virsh si cela n'est pas fait automatiquement :

```
[root@cen7 ~]# virsh start win7  
Domain win7 started
```

Étape 4. Une fois l'installation terminée, arrêtez la machine virtuelle. Vous avez maintenant une
image QCOW de cette installation dans le chemin d'accès fourni lors de la création de la machine
virtuelle : `/var/lib/libvirt/images/win7.img`. Ce type d'image peut être déployé sur CGM-SRV pour
exécuter Windows.

Déployer une image de machine virtuelle Windows sur CGM-SRV

Maintenant que vous avez le type d'image correct à exécuter sur le CGM-SRV, vous pouvez commencer à le déployer.

Étape 1. Configurez un profil pour le client qui correspond à votre configuration :

```
[root@cen7 ~]# ./ioxclient profiles create
Enter a name for this profile : CGR1120_20
Your IOx platform's IP address[127.0.0.1] : 10.X.X.X.X
Your IOx platform's port number[8443] :
Authorized user name[root] : admin
Password for admin :
Local repository path on IOx platform[/software/downloads]:
URL Scheme (http/https) [https]:
API Prefix[/iox/api/v2/hosting/]:
Your IOx platform's SSH Port[2222]:
Your RSA key, for signing packages, in PEM format[]:
Your x.509 certificate in PEM format[]:
Activating Profile  CGR1120_20
Saving current configuration
```

Dans cet exemple, 10.X.X.X correspond à l'interface sortante sur le CGR1000 sur laquelle vous avez configuré la traduction d'adresses de réseau (NAT) pour transférer vers le port 8443 sur le CGM-SRV.

Étape 2. Maintenant que ioxclient est configuré, renommons votre image créée précédemment en **vm.img** afin de simplifier un peu et copions-la avec l'utilisation de Secure Copy (SCP) avec ioxclient en CGM-SRV.

Le cas échéant, convertissez l'image disque au format QCOW2, comme le CGM-SRV l'attend. Les nouvelles versions de virt-manager semblent créer les images de disque par défaut au format QCOW3.

Vous pouvez facilement convertir l'image à l'aide de cette commande :

```
[root@cen7 ~]# qemu-img convert -f qcow2 -O qcow2 /var/lib/libvirt/images/win7.img
/var/lib/libvirt/images/win7.img
```

Une fois que vous êtes sûr que l'image est au bon format, poursuivez avec le nom et la copie :

```
[root@cen7 ~]# mv /var/lib/libvirt/images/win7.img /root/vm.img
[root@cen7 ~]# ./ioxclient platform scp /root/vm.img
Currently active profile :  CGR1120_20
Command Name:  plt-scp
Saving current configuration
Downloaded scp keys to pscp.pem
Running command : [scp -P 2222 -r -i pscp.pem /root/vm.img scpuser@10.50.215.246:/]
```

Ce transfert peut prendre un certain temps, les débits de transfert d'environ 3-4 Mo/s à CGM-SRV via Cisco IOS®. Le fichier est copié sur **/mnt/data/vm/vm.img** sur le module CGM-SRV.

Étape 3. Lorsque le transfert est en cours (ou terminé), vous pouvez créer le fichier **package.yaml**. Ce fichier décrit à IOx ce que vous souhaitez déployer exactement et comment l'empaqueter.

```
[root@cen7 ~]# vi package.yaml
[root@cen7 ~]# cat package.yaml
descriptor-schema-version: 2.2
```

info:

```
author-link: http://www.cisco.com/ author-name: Jens Depuydt description: Windows 7 VM for
CSR-SRV name: win7 version: 1.0 app: type: vm cpuarch: x86_64 resources: profile: custom cpu:
600 disk: 10 memory: 3072 network: - interface-name: eth0 - interface-name: eth1 graphics: vnc:
true startup: ostype: windows qemu-guest-agent: false disks: - target-dev: hda file:
file://vm.img
```

Comme vous pouvez le voir dans ce **package.yaml**, vous référez à **file://vm.img** qui correspond à l'emplacement réel de **mnt/data/vm/vm.img** sur le module CGM-SRV.

Étape 4. L'étape suivante consiste à empaqueter avec l'utilisation d'ioxclient :

```
[root@cen7 ~]# ./ioxclient pkg .
Currently active profile : default
Command Name: package
No rsa key and/or certificate files to sign the package
Checking if package descriptor file is present..
Validating descriptor file /root/package.yaml with package schema definitions
Parsing descriptor file..
Found schema version 2.2
Loading schema file for version 2.2
Validating package descriptor file..
File /root/package.yaml is valid under schema version 2.2
Created Staging directory at : /var/folders/sp/f9qn2fsn0d5fkj7szps6qvvr0000gn/T/638513626
Copying contents to staging directory
Checking for application runtime type
Couldn't detect application runtime type
Creating an inner envelope for application artifacts
Excluding .DS_Store
Generated /var/folders/sp/f9qn2fsn0d5fkj7szps6qvvr0000gn/T/638513626/artifacts.tar.gz
Calculating SHA1 checksum for package contents..
Package MetaData file was not found at
/private/var/folders/sp/f9qn2fsn0d5fkj7szps6qvvr0000gn/T/638513626/.package.metadata
Wrote package metadata file :
/private/var/folders/sp/f9qn2fsn0d5fkj7szps6qvvr0000gn/T/638513626/.package.metadata
Root Directory : /private/var/folders/sp/f9qn2fsn0d5fkj7szps6qvvr0000gn/T/638513626
Output file: /var/folders/sp/f9qn2fsn0d5fkj7szps6qvvr0000gn/T/559089521
Path: .package.metadata
SHA1 : 262f763740c182f95358be84514a76ac11e37012
Path: artifacts.tar.gz
SHA1 : 3d89ccd35fe5318dd83a249a26cb8140d98d15bb
Path: package.yaml
SHA1 : aa42f949b707df07a83a17344e488c44eb585561
Generated package manifest at package.mf
Generating IOx Package..
Package generated at /root/package.tar
```

Étape 5. Après avoir créé le paquet, vous pouvez l'installer sur notre CGM-SRV. L'application/la machine virtuelle IOx est appelée win7 dans cet exemple :

```
[root@cen7 ~]# ./ioxclient app install win7 package.tar
Currently active profile : default
Command Name: application-install
Saving current configuration
```

Installation Successful. App is available at :

<https://10.X.X.X:8443/iox/api/v2/hosting/apps/win7> Successfully deployed

Étape 6. Avant de pouvoir activer la machine virtuelle win7 IOx, vous devez créer un fichier JSON de charge utile qui définit le mot de passe VNC pour cette machine virtuelle :

```
[root@cen7 ~]# vi vnc.json
[root@cen7 ~]# cat vnc.json
{
  "resources": {
    "graphics": {"vnc-password": "password"}
  }
}
```

Étape 7. Avec l'utilisation de la charge utile **vnc.json**, vous pouvez activer la machine virtuelle win7 IOx :

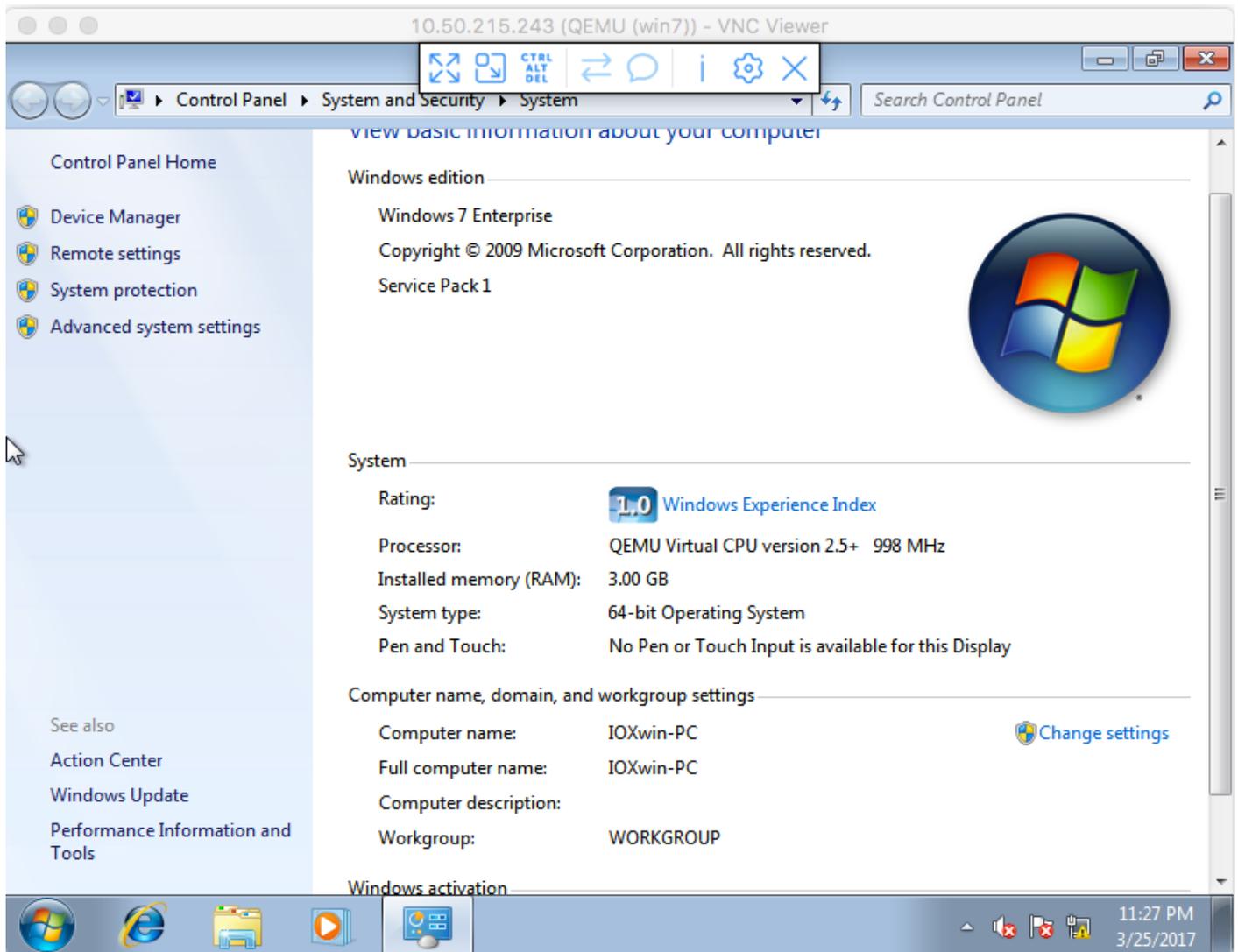
```
[root@cen7 ~]# ./ioxclient app activate win7 --payload vnc.json
Currently active profile : default
Command Name: application-activate
Payload file : vnc.json. Will pass it as application/json in request body..
App win7 is Activated
```

Étape 8. La dernière étape avec ioxclient consiste à démarrer la machine virtuelle :

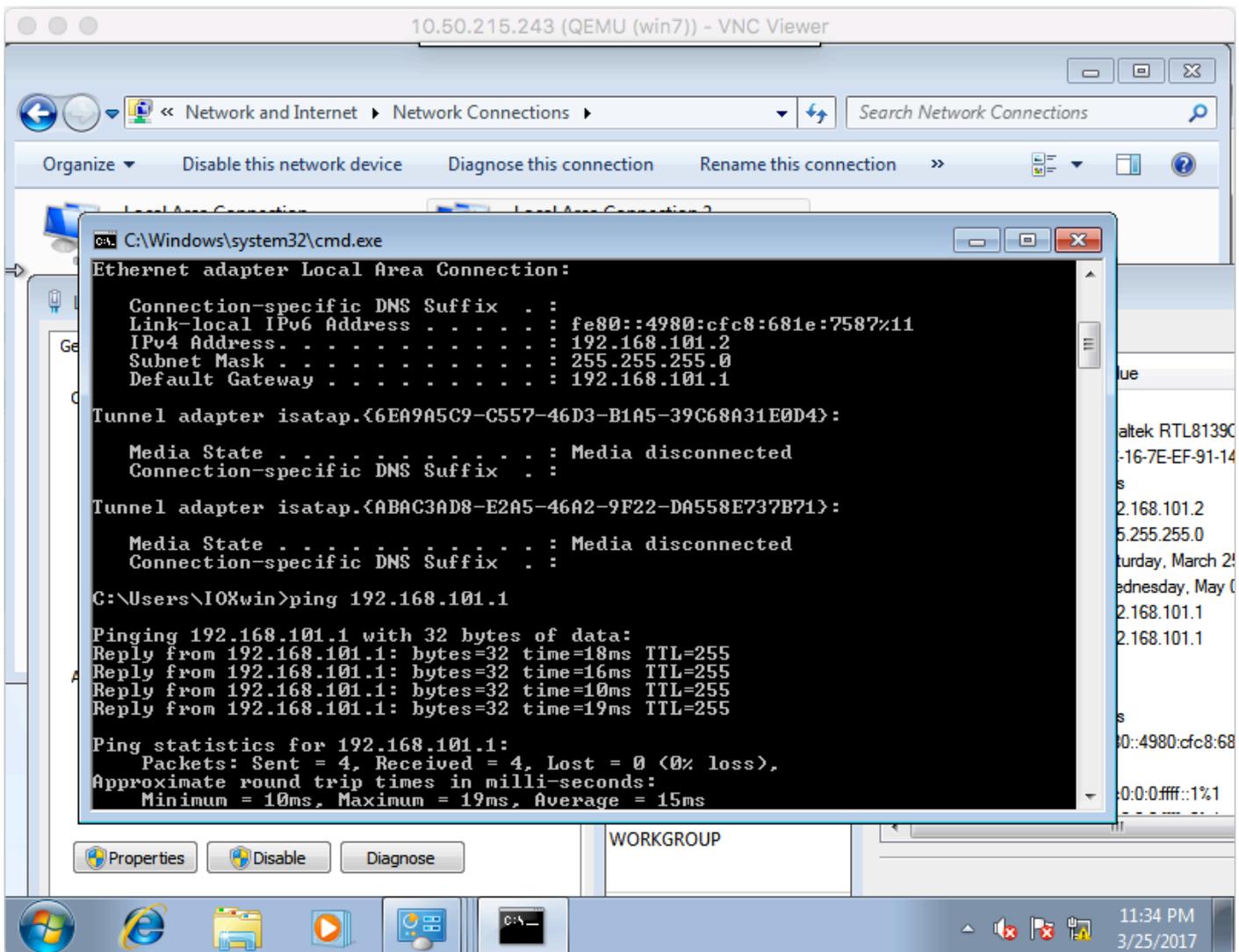
```
[root@cen7 ~]# ./ioxclient app start win7
Currently active profile : default
Command Name: application-start
App win7 is Started
```

À ce stade, la machine virtuelle Windows s'exécute sur CGM-SRV et vous pouvez commencer à l'utiliser.

Afin d'accéder à la console de l'ordinateur Windows, vous pouvez utiliser la visionneuse VNC sur l'interface sortante sur le CGR1000 et le port 5900, comme illustré dans l'image :



Du point de vue du réseau, vous avez choisi de donner eth0 et eth1 à la machine virtuelle d'E/Sx win7 avec l'utilisation du fichier **package.yaml** comme illustré dans l'image :



Comme vous pouvez le voir, ces interfaces ont reçu une adresse IP du serveur DHCP qui fonctionne sur Cisco IOS® et peut être utilisée sans configuration supplémentaire.

Vérification

Utilisez cette section pour confirmer que votre configuration fonctionne correctement.

Afin de vérifier si la machine virtuelle s'exécute :

```
[root@cen7 ~]# ./ioxclient app list
Currently active profile : CGR1120_20
Command Name: application-list
Saving current configuration
List of installed App :
 1. win7      --->  RUNNING
```

Vous pouvez également vérifier l'état à partir du Gestionnaire local, comme indiqué dans l'image :

Application information

| | |
|--------------------|---|
| ID: | win7 |
| State: | RUNNING |
| Name: | win7 |
| Cartidge Required: | * None |
| Version: | 1.0 |
| Author: | Jens Depuydt |
| Author link: | http://www.cisco.com/ |
| Application type: | vm |
| Description: | Windows 7 VM for CSR-SRV |

Requested Resource

| | |
|----------|---------------|
| Cpu: | 600 cpu-units |
| Memory: | 3072 MB |
| Profile: | custom |
| Disk: | 10 MB |
| Vcpu: | 1 |

Network information

| | |
|-----------------|---|
| interface-name: | eth0 eth1 |
|-----------------|---|

App Access

| | |
|----------------|---|
| Console Access | <code>ssh -p {SSH_PORT} -i win7.pem appconsole@10.50.215.243</code> |
| VNC Access | VNC password :password |

Dépannage

Cette section fournit des informations que vous pouvez utiliser pour dépanner votre configuration.

Afin de résoudre les problèmes de déploiement, vérifiez le résultat de `ioxclient` ou `/var/log/caf.log` sur le système d'exploitation hôte CGM-SRV.

Assurez-vous que NAT est configuré correctement pour accéder à toutes les ressources (Cisco Application-hosting Framework (CAF), Secure Shell (SSH), VNC).