

Configurazione di Windows VM al modulo CGM-SRV su CGR1xxx

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Configurazione](#)

[Crea immagine VM Windows](#)

[Installazione di KVM su computer Linux](#)

[Verifica dell'installazione KVM](#)

[Creare una VM Windows](#)

[Distribuire l'immagine VM di Windows in CGM-SRV](#)

[Verifica](#)

[Risoluzione dei problemi](#)

Introduzione

In questo documento vengono descritti i passaggi necessari per creare ed eseguire una macchina virtuale Windows (VM) nel modulo Connected Grid Module (CGM) - System Server (SRV).

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Linux
- Macchina virtuale basata su kernel (KVM)
- Comprendere i concetti della virtualizzazione

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- Connected Grid Router (CGR) 1120
- Modulo CGM-SRV-XX
- I passaggi di configurazione per CGM-SRV vengono eseguiti prima di questa guida:
- ISO di installazione di Windows 7
- Visualizzatore VNC (Virtual Network Computing)

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Per eseguire applicazioni IOx o VM sulla piattaforma CGR1000, è possibile utilizzare il modulo di calcolo CGM-SRV. Il modulo CGM-SRV è in realtà un piccolo server che contiene una CPU x86 multi-core, memoria e storage. Sia CGR1120 che CGR1240 possono includere uno di questi moduli per aggiungere funzionalità IOx.

Al momento della stesura del presente documento sono disponibili due tipi di file:

SKU (Stock Keep Unit)	Unità a stato solido (SSD)	RAM	CPU
CGM-SRV-64	64 GB (50 GB utilizzabili)	4 GB	4 core da 800 MHz
CGM-SRV-128	128 GB (100 GB utilizzabili)	4 GB	4 core da 800 MHz

Ogni modulo è inoltre dotato di due porte USB per lo storage e di una propria interfaccia Gigabit Ethernet esterna.

Come qualsiasi altro dispositivo compatibile con IOx, il modulo può ospitare diversi tipi di applicazioni IOx ma, a causa della maggiore capacità del modulo CGM-SRV, può anche eseguire una distribuzione Windows o Linux standard completamente configurata (ad esempio Ubuntu o CentOS).

Configurazione

Crea immagine VM Windows

Per distribuire una VM Windows sul modulo CGM-SRV, è necessario innanzitutto creare un'immagine in formato QEMU QCOW contenente l'installazione di Windows. Un modo per creare un'immagine di questo tipo consiste nell'utilizzare KVM e virsh su un computer Linux.

Le operazioni descritte più avanti non riguardano affatto il CGR1xxx o il CGM-SRV, sono solo passaggi necessari per creare un'immagine QCOW di Windows 7 VM di base che è possibile installare nella fase successiva del CGM-SRV.

Per questa guida, è possibile iniziare con una nuova installazione minima di CentOS7. I passaggi per le altre distribuzioni Linux devono essere simili, ma possono essere leggermente diversi.

Installazione di KVM su computer Linux

Passaggio 1. È innanzitutto necessario verificare se il computer host supporta le estensioni VM. Sulla piattaforma x86, questi sono AMD-V o Intel VT-X. La maggior parte, se non tutte, le moderne CPU x86 supportano queste estensioni. Anche quando si esegue una VM, la maggior parte degli hypervisor fornisce l'opzione per passare/emulare queste estensioni.

Per verificare se la CPU installata supporta tali estensioni, è necessario verificare se il flag vmx (per VT-X) o svm (per AMD-V) esiste nell'output `cpuinfo`.

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

2

Se l'output di questo comando è 0, significa che nessuna CPU trovata supporta le estensioni della VM. In tal caso, è possibile verificare se tali estensioni sono abilitate nel BIOS o nell'hypervisor quando si utilizza una VM per eseguire questo computer.

Passaggio 2. Il passaggio successivo consiste nella creazione di un bridge per fornire una rete per la macchina virtuale che è possibile eseguire su KVM.

Innanzitutto, è necessario abilitare gli inoltri IP nel kernel:

```
[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
```

Per creare il bridge, la configurazione IP deve spostarsi dall'interfaccia reale al bridge stesso, in quanto questa è l'interfaccia a cui appartiene l'indirizzo IP.

Dopo aver completato un'installazione standard, la configurazione di rete è in **/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
```

Passaggio 3. Come si può vedere, esiste attualmente un'interfaccia (oltre all'interfaccia di loopback), chiamata eno16777736. È necessario spostare la configurazione relativa all'IP su un'interfaccia bridge che può essere chiamata 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
```

Passaggio 4. Quindi, pulire la configurazione IP dall'interfaccia reale e collegarla al bridge 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
```

Passaggio 5. Una volta completata la configurazione della rete, è possibile procedere all'installazione di KVM:

```
[root@cen7 ~]# sudo yum install kvm virt-manager libvirt virt-install qemu-kvm xauth dejavu-lgc-
```

```
sans-fonts -y
```

```
...
```

```
Complete!
```

Passaggio 6. Al termine dell'installazione, la soluzione migliore consiste nel riavviare il computer per applicare i moduli appena installati e la configurazione di rete:

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

Verifica dell'installazione KVM

Passaggio 7. Al termine del riavvio, dovrebbe essere possibile accedere al computer con lo stesso indirizzo IP configurato sull'interfaccia del bridge. È necessario verificare se il modulo del kernel KVM è caricato:

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

Passaggio 8. Se il problema persiste, è possibile provare a connettersi a virsh:

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

Passaggio 9. Un ultimo passaggio consiste nell'aprire la porta 5900 sul firewall del computer per l'accesso VNC all'installazione di Windows:

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

Creare una VM Windows

Ora che si dispone di un sistema che funziona con l'installazione KVM, è possibile attivare una nuova VM su KVM ed eseguire attraverso le finestre di dialogo di installazione di Windows.

Passaggio 1. Copiare l'ISO di installazione di Windows 7 nella macchina virtuale (o renderlo accessibile in rete):

```
[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
```

Passaggio 2. Creare una nuova VM KVM e avviarla dall'ISO di 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

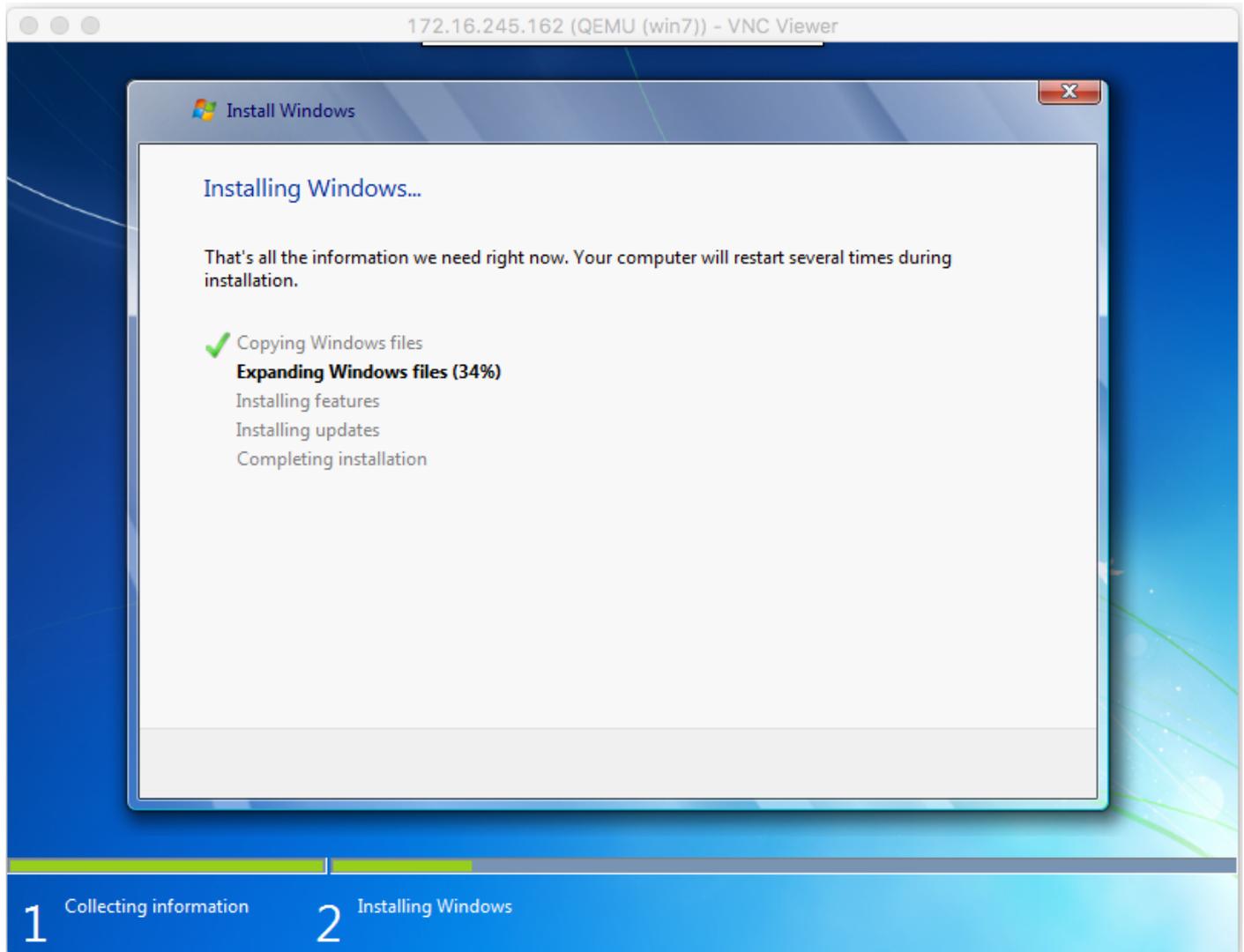
Creating domain...

| 0 B

00:00:00

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

Passaggio 3. Una volta avviata la VM, è possibile connettersi con l'utilizzo del visualizzatore VNC all'indirizzo IP del computer host sulla porta 5900 e completare l'installazione standard di Windows come mostrato nell'immagine:



Se Windows viene riavviato al momento dell'installazione, potrebbe essere necessario riavviare la VM con virsh se questa operazione non viene eseguita automaticamente:

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

Passaggio 4. Al termine dell'installazione, arrestare la VM. È ora disponibile un'immagine QCOW di questa installazione nel percorso fornito al momento della creazione della VM: `/var/lib/libvirt/images/win7.img`. Questo tipo di immagine può essere implementato sul CGM-SRV per eseguire Windows.

Distribuire l'immagine VM di Windows in CGM-SRV

Ora che si dispone del tipo corretto di immagine da eseguire su CGM-SRV, è possibile iniziare a distribuirla.

Passaggio 1. Impostare un profilo per ioxclient corrispondente alla configurazione:

```
[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
```

Nell'esempio, 10.X.X.X corrisponde all'interfaccia in uscita del CGR1000 su cui Network Address Translation (NAT) è stato configurato per l'inoltro alla porta 8443 sul CGM-SRV.

Passaggio 2. Ora che ioxclient è configurato, rinominare l'immagine creata in precedenza in **vm.img** per semplificare un bit e copiarla con l'uso di Secure Copy (SCP) con ioxclient in CGM-SRV.

Opzionalmente, convertire l'immagine del disco nel formato QCOW2 come previsto da CGM-SRV. Le versioni più recenti di virt-manager sembrano creare le immagini disco per impostazione predefinita nel formato QCOW3.

È possibile convertire facilmente l'immagine utilizzando questo comando:

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

Dopo aver verificato che il formato dell'immagine sia corretto, procedere con la ridenominazione e la copia:

```
[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:/]
```

Il trasferimento potrebbe richiedere alcuni minuti, con velocità di trasferimento da circa 3-4 MB/s al CGM-SRV attraverso Cisco IOS®. Il file viene copiato in **/mnt/data/vm/vm.img** sul modulo CGM-SRV.

Passaggio 3. Mentre il trasferimento è in corso (o completato), è possibile creare il file **package.yaml**. In questo file viene descritto a IOx cosa si desidera distribuire e come creare il pacchetto.

```
[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
```

Come si può vedere in questo **package.yaml**, si fa riferimento a **file://vm.img**, che corrisponde alla posizione reale di **mnt/data/vm/vm.img** sul modulo CGM-SRV.

Passaggio 4. Il passaggio successivo consiste nel creare un pacchetto con l'utilizzo di **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
```

Passaggio 5. Dopo aver creato il pacchetto, è possibile installarlo sul nostro CGM-SRV. In questo esempio, l'applicazione/macchina virtuale IOx è denominata win7:

```
[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
```

Passaggio 6. Prima di poter attivare la macchina virtuale Win7 IOx, è necessario creare un file JSON di payload che imposti la password VNC per questa macchina virtuale:

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

Passaggio 7. Utilizzando il payload **vnc.json**, è possibile attivare la VM Win7 IOx:

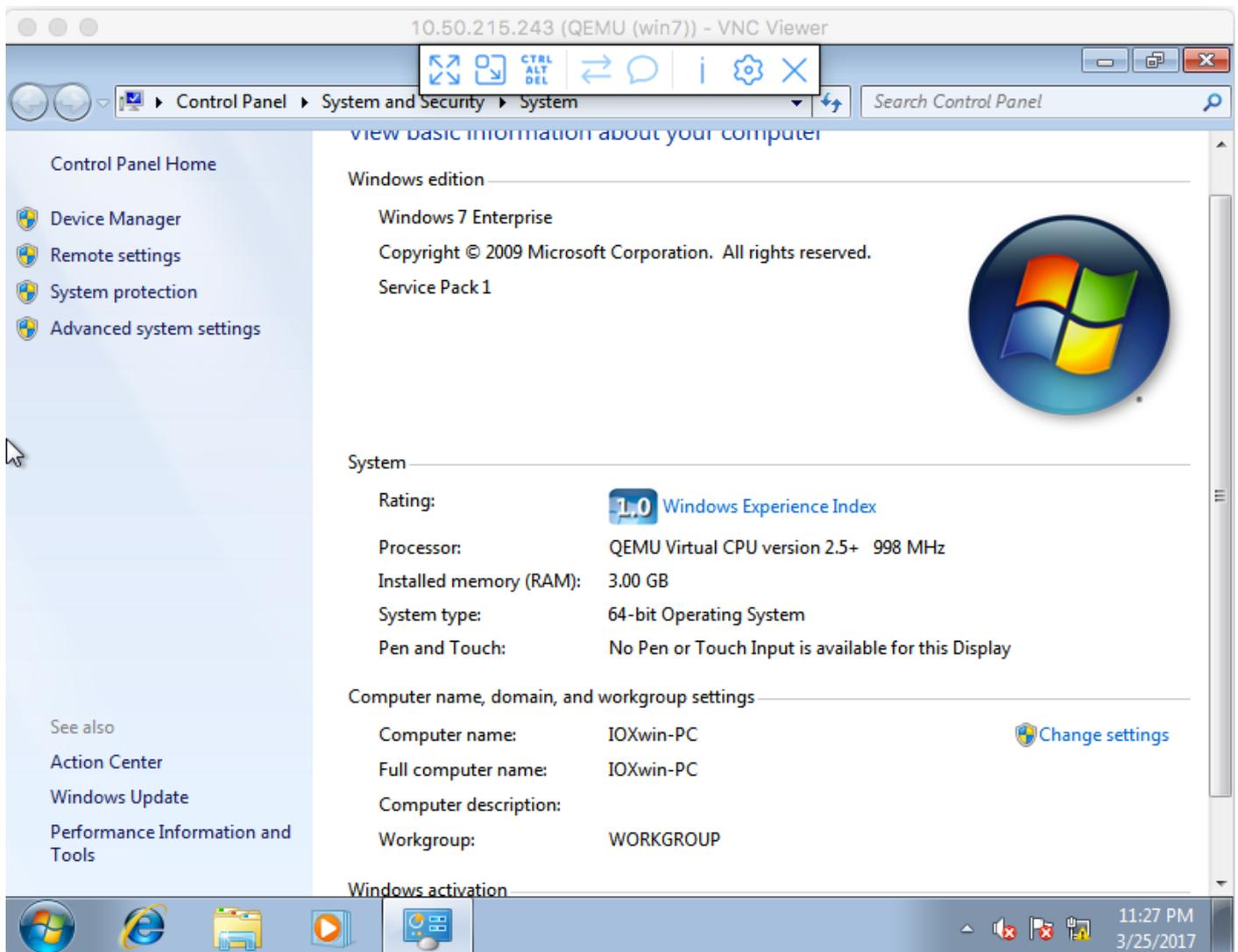
```
[root@cen7 ~]# ./ioxclient app activate win7 --payload vnc.json
Currently active profile : default
Command Name: application-activate
Payload file : vnc.json. Will pass it as application/json in request body..
App win7 is Activated
```

Passaggio 8. L'ultimo passaggio con ioxclient consiste nell'avviare la macchina virtuale:

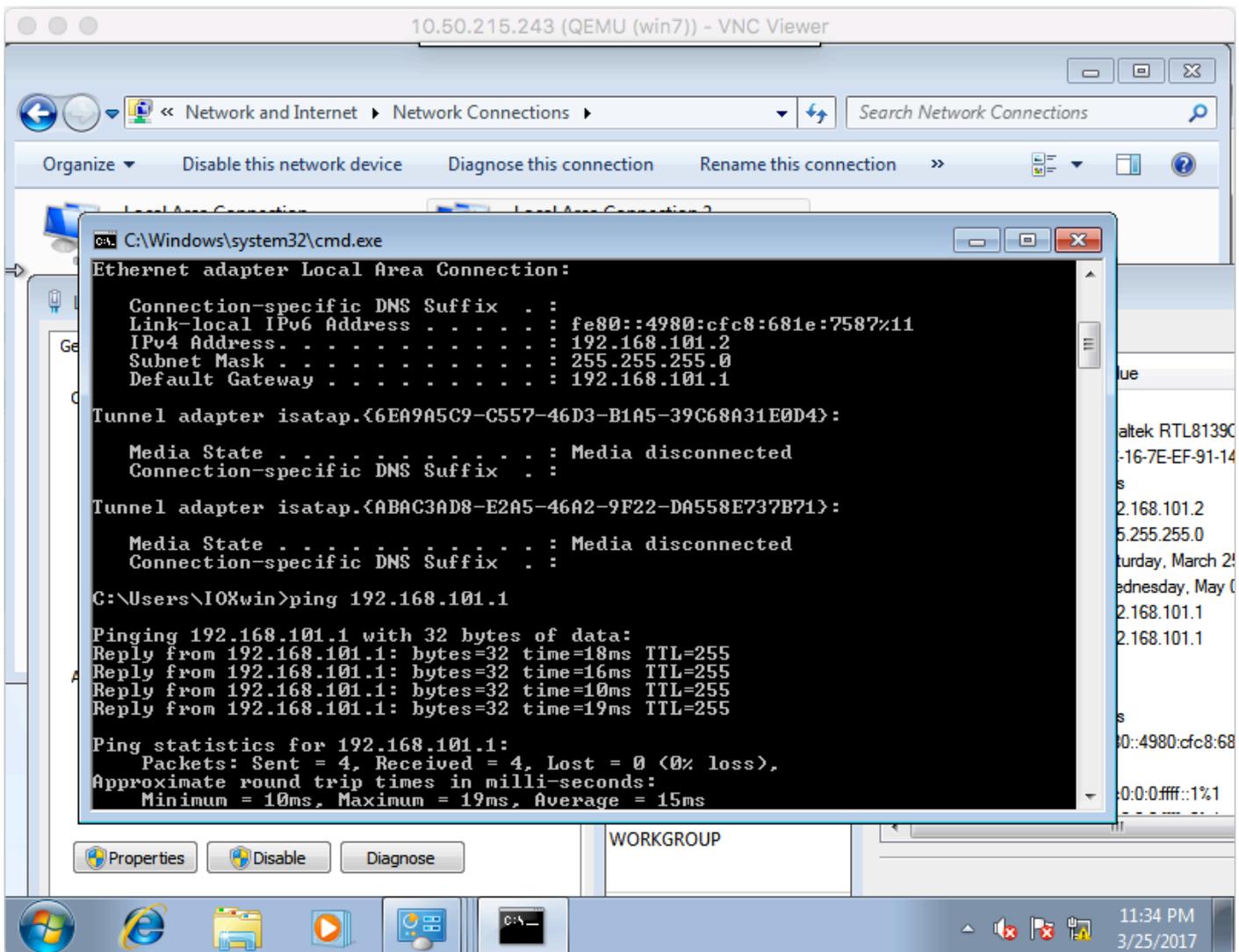
```
[root@cen7 ~]# ./ioxclient app start win7
Currently active profile : default
Command Name: application-start
App win7 is Started
```

A questo punto, la VM Windows viene eseguita su CGM-SRV e si può iniziare a utilizzarla.

Per accedere alla console del computer Windows, è possibile utilizzare il visualizzatore VNC sull'interfaccia in uscita del CGR1000 e sulla porta 5900, come mostrato nell'immagine:



Dal punto di vista della rete, si è scelto di assegnare eth0 e eth1 alla VM Win7 IOx con l'uso del file `package.yaml` come mostrato nell'immagine:



Come si può vedere, queste interfacce hanno un indirizzo IP dal server DHCP che funziona su Cisco IOS® e può essere usato senza ulteriori configurazioni.

Verifica

Fare riferimento a questa sezione per verificare che la configurazione funzioni correttamente.

Per verificare se la VM è in esecuzione:

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

È inoltre possibile controllare lo stato da Gestione locale, come mostrato nell'immagine:

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

Risoluzione dei problemi

Le informazioni contenute in questa sezione permettono di risolvere i problemi relativi alla configurazione.

Per risolvere i problemi relativi all'installazione, controllare l'output di `ioxclient` o `/var/log/caf.log` sul sistema operativo host CGM-SRV.

Verificare che NAT sia configurato correttamente per accedere a tutte le risorse (Cisco Application-Hosting Framework (CAF), Secure Shell (SSH), VNC).