

CGR1xxxのCGM-SRVモジュールへのWindows VMの設定

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[設定](#)

[Windows VMイメージの作成](#)

[LinuxマシンへのKVMのインストール](#)

[KVMのインストールの確認](#)

[Windows VMの作成](#)

[Windows VMイメージのCGM-SRVへの展開](#)

[確認](#)

[トラブルシューティング](#)

概要

このドキュメントでは、Connected Grid Module(CGM) - System Server(SRV)モジュールでWindows仮想マシン(VM)を作成して実行するために必要な手順について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Linux
- カーネルベースの仮想マシン(KVM)
- 仮想化の概念の理解

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Connected Gridルータ(CGR)1120
- CGM-SRV-XXモジュール
- CGM-SRVの設定手順は、このガイドの前に実行してください。
- Windows 7インストールISO
- Virtual Network Computing(VNC)ビューア

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このド

キュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

CGR1000プラットフォームでIOxアプリケーションまたはVMを実行する場合は、CGM-SRVコンピュティングモジュールを使用できます。CGM-SRVモジュールは、実際にはマルチコアx86 CPU、メモリ、ストレージを搭載した小型サーバです。CGR1120とCGR1240の両方に、IOx機能を追加するためにこれらのモジュールのいずれかを搭載できます。

執筆時点で、次の2つのタイプを使用できます。

在庫保持ユニット(SKU)	ソリッドステートドライブ(SSD)	RAM	CPU
CGM-SRV-64	64 GB (50 GB使用可能)	4GB	4コア800 Mhz
CGM-SRV-128	128 GB (100 GB使用可能)	4GB	4コア800 Mhz

各モジュールには、ストレージ用の2つのUSBポートと、独自の外部ギガビットイーサネットインターフェイスがあります。

他のIOx対応デバイスと同様に、このモジュールはさまざまなタイプのIOxアプリケーションをホストできますが、CGM-SRVモジュールの容量が大きいため、Windowsまたは標準のLinuxディストリビューション (UbuntuやCentOSなど) を実行できます。

設定

Windows VMイメージの作成

CGM-SRVモジュールにWindows VMを展開するには、まず、Windowsインストールを含むQEMU QCOW形式のイメージを作成する必要があります。このようなイメージを作成する1つの方法は、Linuxマシン上でKVMと仮想コマンドを使用することです。

さらに説明する手順には、CGR1xxxやCGM-SRVは一切含まれていません。これらは、CGM-SRVの次の手順で展開できる基本的なWindows 7 VM QCOWイメージを作成するために必要な手順です。

このガイドでは、CentOS7の最小限のインストールから開始できます。他のLinuxディストリビューションの手順は同じでなければなりません、少し異なる場合があります。

LinuxマシンへのKVMのインストール

ステップ1：まず、ホストマシンがVM拡張をサポートしているかどうかを確認します。x86プラットフォームでは、これらはAMD-VまたはIntelのVT-Xです。ほとんどの場合、最新のx86 CPUはこれらの拡張をサポートしています。VMを実行する場合でも、ほとんどのハイパーバイザには、これらの拡張機能を通過/エミュレートするオプションがあります。

インストールされたCPUがこれらの拡張をサポートしているかどうかを確認するには、`cpuinfo-output`に`vmx` (VT-X用) または`svm` (AMD-V用) フラグが存在するかどうかを確認する必要があります。

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

2

このコマンドの出力が0の場合、CPUが見つからず、VM拡張をサポートしていることを意味します。この場合、VMを使用してこのマシンを実行するときに、BIOSまたはハイパーバイザでこれらの拡張機能が有効になっているかどうかを確認できます。

ステップ2：次のステップは、KVMで実行できるVMのネットワークを提供するブリッジを作成することです。

まず、カーネルでIP転送を有効にする必要があります。

```
[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
```

ブリッジを作成するには、IP設定を実際のインターフェイスからブリッジ自体に移動する必要があります。これは、IPアドレスを所有するインターフェイスであるためです。

標準インストールが完了すると、ネットワーク構成は/etc/sysconfig/network-scriptsになります。

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

```
/etc/sysconfig/network-scripts/ifcfg-eno16777736
```

```
/etc/sysconfig/network-scripts/ifcfg-lo
```

ステップ3：現在、eno16777736というインターフェイスが1つあります（ループバックインターフェイス以外にも）。IP関連の設定を、virbr0を呼び出すことができるブリッジインターフェイスに移動する必要があります。

```
[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
```

ステップ4：その後、実際のインターフェイスからIP設定をクリーンアップし、それをvirbr0ブリッジに接続する必要があります。

```
[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
```

ステップ5：ネットワーク設定が完了したら、先に進んでKVMをインストールします。

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

...

Complete!

ステップ6：インストールが完了したら、このマシンをリブートして、新しくインストールされたモジュールとネットワーク設定を適用するのが最適です。

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

KVMのインストールの確認

ステップ7：リブートが完了すると、ブリッジインターフェイスで設定されている（同じ）IP上のマシンにアクセスできるようになります。KVMカーネルモジュールがロードされているかどうかを確認する必要があります。

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

ステップ8：問題が解決しない場合は、次のようにvirshを使用して接続を試すことができます。

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

ステップ9：最後の1つのステップは、WindowsインストールにVNCアクセスするためにこのマシンのファイアウォールのポート5900を開くことです。

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

Windows VMの作成

KVMインストールで動作するシステムが完成したので、KVMで新しいVMを起動し、Windowsのインストール・ダイアログを実行できます。

ステップ1:Windows 7インストールISOをVMにコピーします（またはネットワーク経由でアクセス可能にします）。

```
[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
```

ステップ2：新しいKVM VMを作成し、Windows 7 ISOから起動させます。

```
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
00:00:00
```

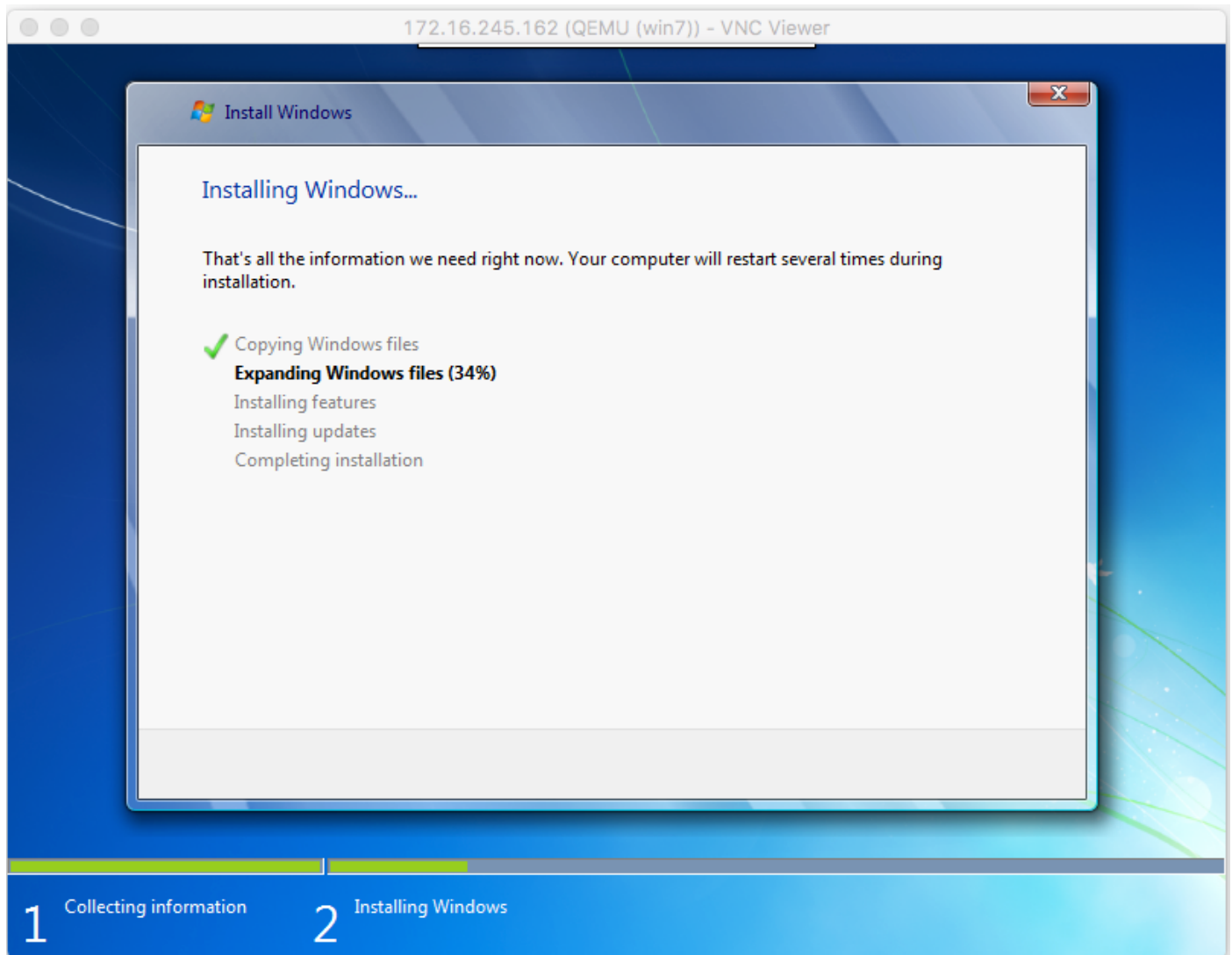
| 9.0 GB

Creating domain...
00:00:00

| 0 B

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

ステップ3:VMが起動したら、VNCビューアを使用してポート5900のホストマシンのIPに接続し、図に示すように標準のWindowsインストールを終了します。



インストール時にWindowsがリブートする場合、これが自動的に行われない場合は、virshを使用してVMを再起動する必要があります。

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

ステップ4: インストールが完了したら、VMをシャットダウンします。これで、VMの作成時に指定されたパスにこのインストールのQCOWイメージが作成されました。

`/var/lib/libvirt/images/win7.img` にアクセスしてください。このタイプのイメージは、CGM-SRV上に展開してWindowsを実行できます。

Windows VMイメージのCGM-SRVへの展開

CGM-SRVで実行するイメージのタイプが正しくなったので、導入を開始できます。

ステップ1: 設定に対応するioxlclientのプロファイルを設定します。

```
[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
```

この例では、10.X.X.Xは、CGM-SRVのポート8443に転送するようにネットワークアドレス変換(NAT)を設定したCGR1000の発信インターフェイスに対応します。

ステップ2:ioxclientが設定されたら、以前に作成したイメージの名前をvm.imgに変更して、ビットを簡略化し、ioxclientを使用したセキュアコピー(SCP)を使用してCGM-SRVにコピーします。

ディスクイメージをQCOW2フォーマットに変換します。これはCGM-SRVが想定しているものです。virt-managerの新しいバージョンでは、デフォルトでQCOW3形式でディスクイメージが作成されているようです。

次のコマンドを使用すると、イメージを簡単に変換できます。

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

イメージの形式が正しいことを確認したら、名前の変更とコピーに進みます。

```
[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:/]
```

この転送には、Cisco IOS®経由で約3 ~ 4MB/sからCGM-SRVへの転送レートが必要です。ファイルは、CGM-SRVモジュールの/mnt/data/vm/vm.imgにコピーされます。

ステップ3 : 転送が進行中 (または完了) の間に、package.yamlファイルを作成できます。このファイルは、IOxに対して導入する内容とパッケージ方法について説明します。

```
[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
```

このpackage.yamlで分かるように、CGM-SRVモジュールのmnt/data/vm/vm.imgの実際の場所に対応するfile://vm.imgを参照してください。

ステップ4：次のステップは、ioxclientを使用してパッケージ化することです。

```
[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
```

ステップ5：パッケージを作成したら、CGM-SRVにインストールできます。この例では、IOxアプリケーション/VMはwin7と呼ばれます。

```
[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

ステップ6: win7 IOx VMをアクティブにする前に、このVMのVNCパスワードを設定するペイロードJSONファイルを作成する必要があります。

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

```
"graphics": {"vnc-password": "password"}
}
```

ステップ7:vnc.jsonペイロードを使用して、win7 IOx VMをアクティブ化できます。

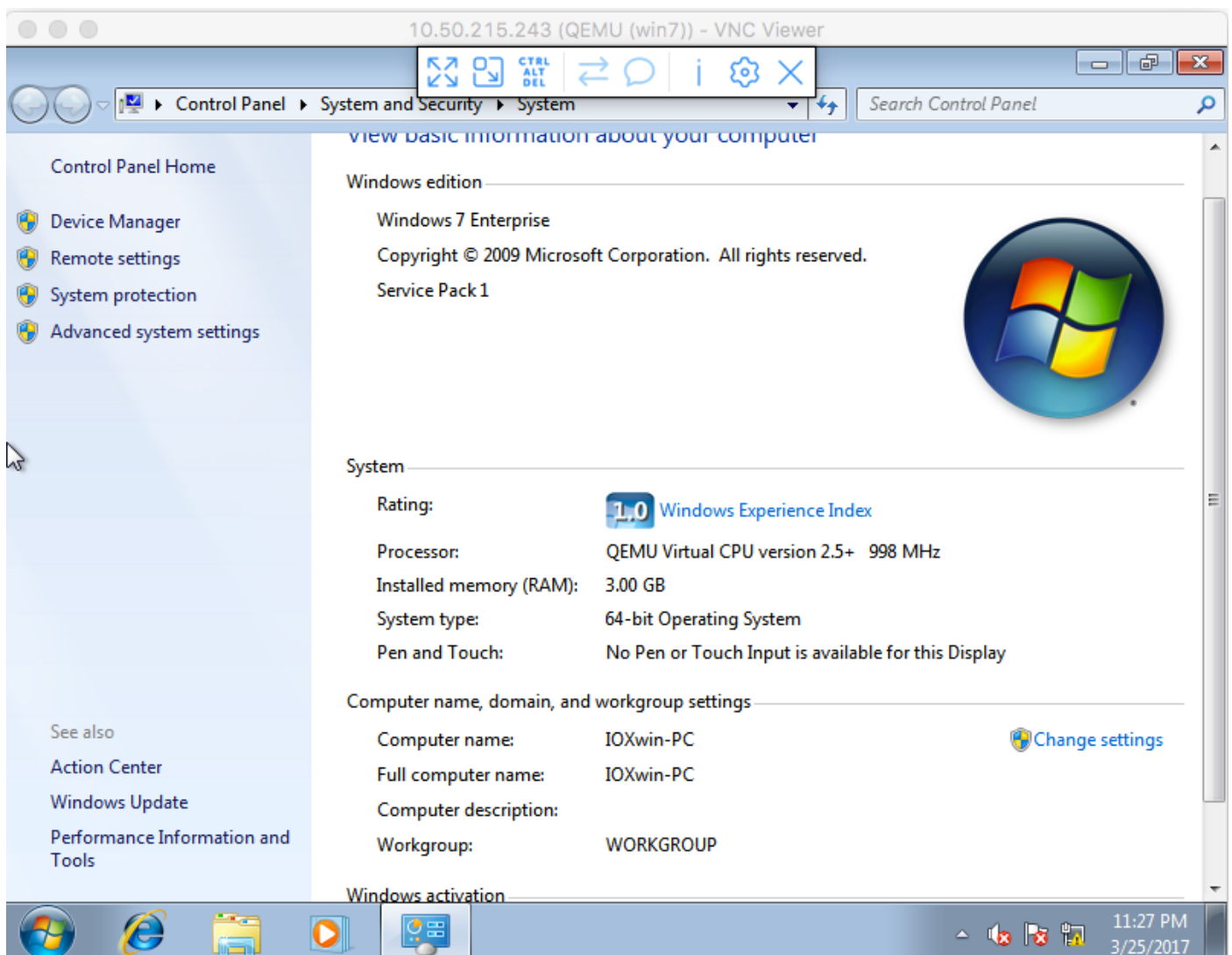
```
[root@cen7 ~]# ./ioxclient app activate win7 --payload vnc.json
Currently active profile : default
Command Name: application-activate
Payload file : vnc.json. Will pass it as application/json in request body..
App win7 is Activated
```

ステップ8:ioxclientの最後のステップは、VMを起動することです。

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

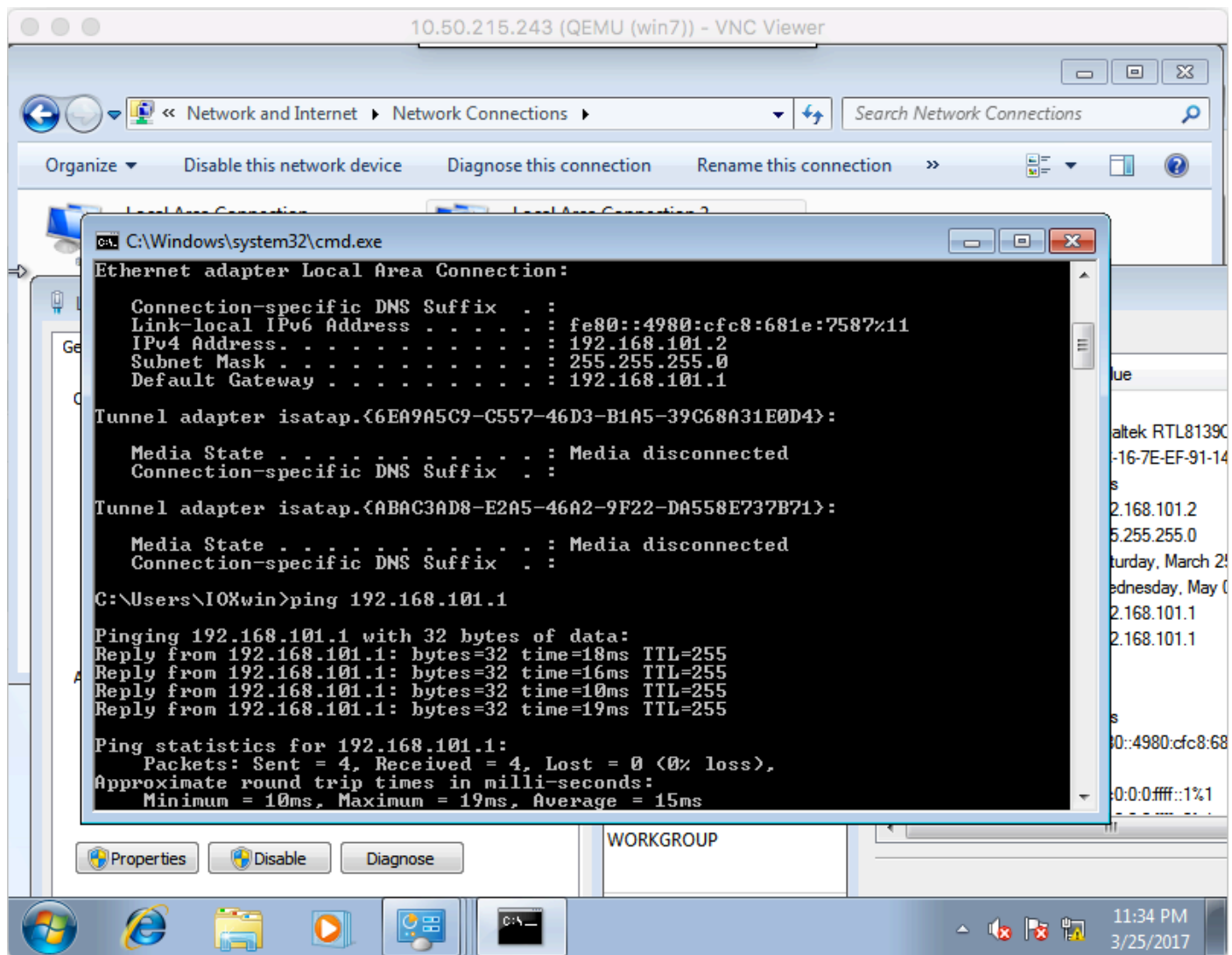
この時点で、Windows VMはCGM-SRV上で実行され、使用を開始できます。

Windowsマシンコンソールにアクセスするには、図に示すように、CGR1000およびポート5900の発信インターフェイスでVNCビューアを使用できます。



ネットワークの観点から、図に示すように、package.yamlファイルを使用して、eth0とeth1を

win7 IOx VMに指定することを選択しました。



これらのインターフェイスは、Cisco IOS®で稼働するDHCPサーバからIPを取得し、それ以上の設定を行わずに使用できます。

確認

ここでは、設定が正常に機能しているかどうかを確認します。

VMが実行されているかどうかを確認するには、次の手順を実行します。

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

図に示すように、Local Managerからステータスを確認することもできます。

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	ssh -p {SSH_PORT} -i win7.pem appconsole@10.50.215.243
VNC Access	VNC password :password

トラブルシューティング

ここでは、設定のトラブルシューティングに使用できる情報を示します。

配備に関する問題をトラブルシューティングするには、CGM-SRVホストOS上のioxclientまたは/var/log/caf.logの出力を確認します。

すべてのリソース(シスコアプリケーションホスティングフレームワーク(CAF)、セキュアシェル(SSH)、VNC)にアクセスするようにNATが正しく設定されていることを確認します。