

CGR1xxx에서 Windows VM을 CGM-SRV 모듈로 구성

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[구성](#)

[Windows VM 이미지 만들기](#)

[Linux 시스템에 KVM 설치](#)

[KVM 설치 확인](#)

[Windows VM 생성](#)

[CGM-SRV에 Windows VM 이미지 배포](#)

[다음을 확인합니다.](#)

[문제 해결](#)

소개

이 문서에서는 CGM(Connected Grid Module) - SRV(System Server) 모듈에서 Windows VM(가상 머신)을 생성하고 실행하는 데 필요한 단계에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Linux
- KVM(Kernel Based Virtual Machine)
- 가상화 개념 이해

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- CGR(Connected Grid Router) 1120
- CGM-SRV-XX 모듈
- CGM-SRV에 대한 컨피그레이션 단계는 이 설명서에 앞서 실행됩니다.
- Windows 7 설치 ISO

- VNC(Virtual Network Computing) 뷰어

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

CGR1000 플랫폼에서 IOx 애플리케이션 또는 VM을 실행하려는 경우 CGM-SRV 컴퓨팅 모듈을 사용할 수 있습니다. CGM-SRV 모듈은 실제로 멀티코어 x86 CPU, 메모리 및 스토리지가 포함된 소규모 서버입니다. CGR1120 및 CGR1240에는 모두 이러한 모듈 중 하나가 장착되어 IOx 기능을 추가할 수 있습니다.

작성 시 사용할 수 있는 두 가지 유형이 있습니다.

재고 유지 장치(SKU)	솔리드 스테이트 드라이브(SSD)	램	CPU
CGM-SRV-64	64GB(가용 용량 50GB)	4GB	4코어 800Mhz
CGM-SRV-128	128GB(가용 용량 100GB)	4GB	4코어 800Mhz

각 모듈에는 스토리지용 USB 포트 2개와 자체 외부 기가비트 이더넷 인터페이스도 있습니다.

다른 IOx 지원 장치와 마찬가지로, 모듈은 다양한 유형의 IOx 애플리케이션을 호스팅할 수 있지만, CGM-SRV 모듈의 용량이 크기 때문에 완전히 구성된 Windows 또는 표준 Linux 디스트로(예: Ubuntu 또는 CentOS)를 실행할 수도 있습니다.

구성

Windows VM 이미지 만들기

CGM-SRV 모듈에 Windows VM을 구축하려면 먼저 Windows 설치가 포함된 QEMU QCOW 형식의 이미지를 생성해야 합니다. 그러한 이미지를 생성하는 한 가지 방법은 Linux 시스템에서 KVM 및 버시를 사용하는 것입니다.

추가로 언급된 단계는 CGR1xxx 또는 CGM-SRV와 전혀 관련이 없으며, 기본 Windows 7 VM QCOW 이미지를 생성하는 데 필요한 단계이며, 다음 단계에서 CGM-SRV에 구축할 수 있습니다.

이 설명서에서는 CentOS7 최소 설치를 새로 설치하는 것으로 시작할 수 있습니다. 다른 Linux 배포판의 단계는 유사해야 하지만 약간 다를 수 있습니다.

Linux 시스템에 KVM 설치

1단계. 먼저 호스트 시스템에서 VM 확장을 지원하는지 확인합니다. x86 플랫폼에서는 AMD-V 또는 Intel VT-X입니다. 전부는 아니지만 대부분의 최신 x86 CPU는 이러한 확장을 지원합니다. VM을 실행할 때도 대부분의 하이퍼바이저는 이러한 확장을 전달/에뮬레이션하는 옵션을 제공합니다.

설치된 CPU에서 이러한 확장을 지원하는지 확인하려면 `cpuinfo-output`에 `vmx`(VT-X용) 또는

svm(AMD-V용) 플래그가 있는지 확인해야 합니다.

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

이 명령의 출력이 0이면 VM 확장을 지원하는 CPU를 찾을 수 없음을 의미합니다. 이 경우 VM을 사용하여 이 시스템을 실행할 때 BIOS 또는 하이퍼바이저에서 이러한 확장이 활성화되었는지 확인할 수 있습니다.

2단계. 다음 단계는 KVM에서 실행할 수 있는 VM에 네트워크를 제공하는 브리지를 생성하는 것입니다.

먼저 커널에서 IP 전달을 활성화해야 합니다.

```
[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
```

브리지를 생성하려면 IP 주소를 소유하는 인터페이스이므로 IP 컨피그레이션을 실제 인터페이스에서 브리지 자체로 이동해야 합니다.

표준 설치를 완료한 후 네트워크 컨피그레이션은 /etc/sysconfig/network-scripts에 있습니다.

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

3단계. 보시다시피 현재 하나의 인터페이스(루프백 인터페이스 외)가 있습니다(eno16777736). IP 관련 컨피그레이션을 virbr0을 호출할 수 있는 브리지 인터페이스로 이동해야 합니다.

```
[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
```

4단계. 그런 다음 실제 인터페이스에서 IP 컨피그레이션을 정리하고 virbr0 브리지에 연결해야 합니다.

```
[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
```

5단계. 네트워크 컨피그레이션이 완료되면 계속해서 KVM을 설치할 수 있습니다.

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

6단계. 설치가 완료되면 이 시스템을 재부팅하여 새로 설치된 모듈 및 네트워크 컨피그레이션을 적용하는 것이 가장 좋습니다.

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

KVM 설치 확인

7단계. 재부팅이 완료되면 브리지 인터페이스에 구성된 동일한 IP에서 시스템에 액세스할 수 있어야 합니다. KVM 커널 모듈이 로드되었는지 확인해야 합니다.

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

8단계. 문제가 해결되지 않으면 virsh에 연결을 시도할 수 있습니다.

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

9단계. 마지막 단계는 Windows 설치에 대한 VNC 액세스를 위해 이 시스템의 방화벽에서 포트 5900을 여는 것입니다.

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

Windows VM 생성

이제 KVM 설치와 함께 작동하는 시스템이 있으므로 KVM에서 새 VM을 시작하고 Windows 설치 대화 상자를 통해 실행할 수 있습니다.

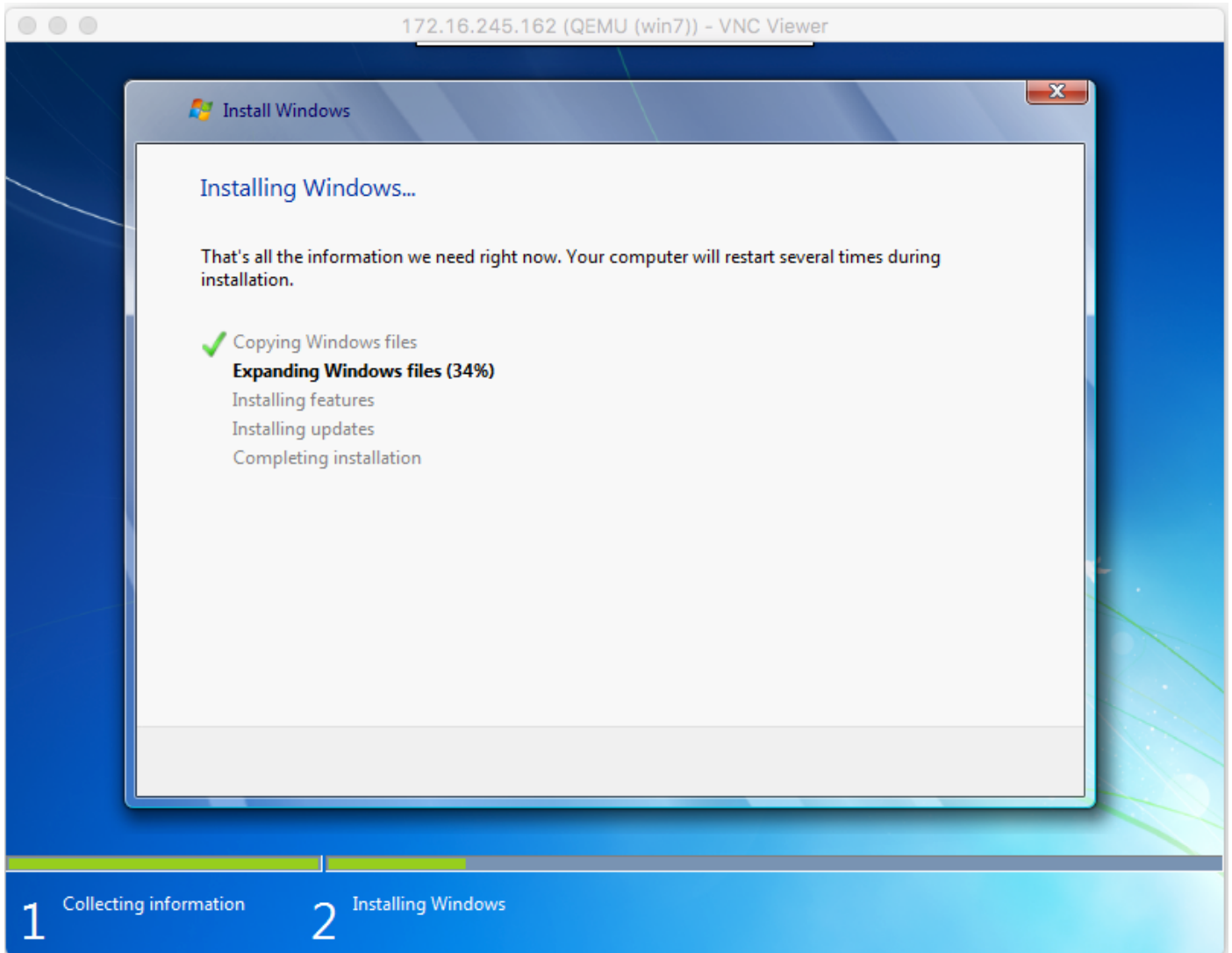
1단계. Windows 7 설치 ISO를 VM에 복사하거나 네트워크를 통해 액세스할 수 있도록 합니다.

```
[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
```

2단계. 새 KVM VM을 생성하고 Windows 7 ISO에서 부팅합니다.

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

3단계. VM이 시작되면 VNC 뷰어를 사용하여 포트 5900의 호스트 머신 IP에 연결한 다음 이미지에 표시된 대로 표준 Windows 설치를 완료할 수 있습니다.



설치 시 Windows가 재부팅되는 경우, 자동으로 실행되지 않으면 버시가 있는 VM을 다시 시작해야 할 수 있습니다.

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

4단계. 설치가 완료되면 VM을 종료합니다. 이제 VM을 생성할 때 제공된 경로에 이 설치의 QCOW 이미지가 있습니다. /var/lib/libvirt/images/win7.img 이 유형의 이미지는 CGM-SRV에 구축하여 Windows를 실행할 수 있습니다.

CGM-SRV에 Windows VM 이미지 배포

이제 CGM-SRV에서 실행할 올바른 유형의 이미지가 있으므로 구축을 시작할 수 있습니다.

1단계. 컨피그레이션에 해당하는 ioxlient에 대한 프로파일을 설정합니다.

<#root>

```
[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
```

이 예에서 10.X.X.X는 CGM-SRV의 포트 8443으로 전달하도록 NAT(Network Address Translation)를 구성한 CGR1000의 발신 인터페이스에 해당합니다.

2단계. ioxclient가 구성되었으므로 비트를 간소화하고 SCP(Secure Copy)를 사용하여 CGM-SRV에 복사하기 위해 먼저 생성한 이미지의 이름을 vm.img로 변경합니다.

선택적으로, 디스크 이미지를 CGM-SRV에 필요한 QCOW2 형식으로 변환합니다. 최신 버전의 virt-manager는 기본적으로 QCOW3 형식으로 디스크 이미지를 만드는 것 같습니다.

다음 명령을 사용하여 이미지를 쉽게 변환할 수 있습니다.

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

이미지가 올바른 형식인지 확인한 후 이름 바꾸기 및 복사를 진행합니다.

<#root>

```
[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:/]
```

이 전송에는 약 3-4MB/s에서 Cisco IOS®를 통해 CGM-SRV로의 전송 속도인 시간이 걸릴 수 있습니다. 파일이 CGM-SRV 모듈의 /mnt/data/vm/vm.img으로 복사됩니다.

3단계. 전송이 진행(또는 완료) 중일 때 package.yaml 파일을 작성할 수 있습니다. 이 파일은 정확히 어떤 것을 배포하고 싶은지 그리고 어떻게 패키징할지 IOx에 대해 설명합니다.

```
[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
```

이 package.yaml에서 볼 수 있듯이, CGM-SRV 모듈에서 mnt/data/vm/vm.img의 실제 위치에 해당하는 file://vm.img을 참조하십시오.

4단계. 다음 단계는 ioxclient를 사용하여 패키징하는 것입니다.

<#root>

```
[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
```

5단계. 패키지를 생성한 후 CGM-SRV에 설치할 수 있습니다. 이 예에서는 IOx 애플리케이션/VM을 win7이라고 합니다.

<#root>

```
[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

6단계. win7 IOx VM을 활성화하기 전에 이 VM에 대한 VNC 비밀번호를 설정하는 페이로드 JSON 파일을 생성해야 합니다.

<#root>

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

7단계. vnc.json 페이로드를 사용하여 win7 IOx VM을 활성화할 수 있습니다.

```
[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
```

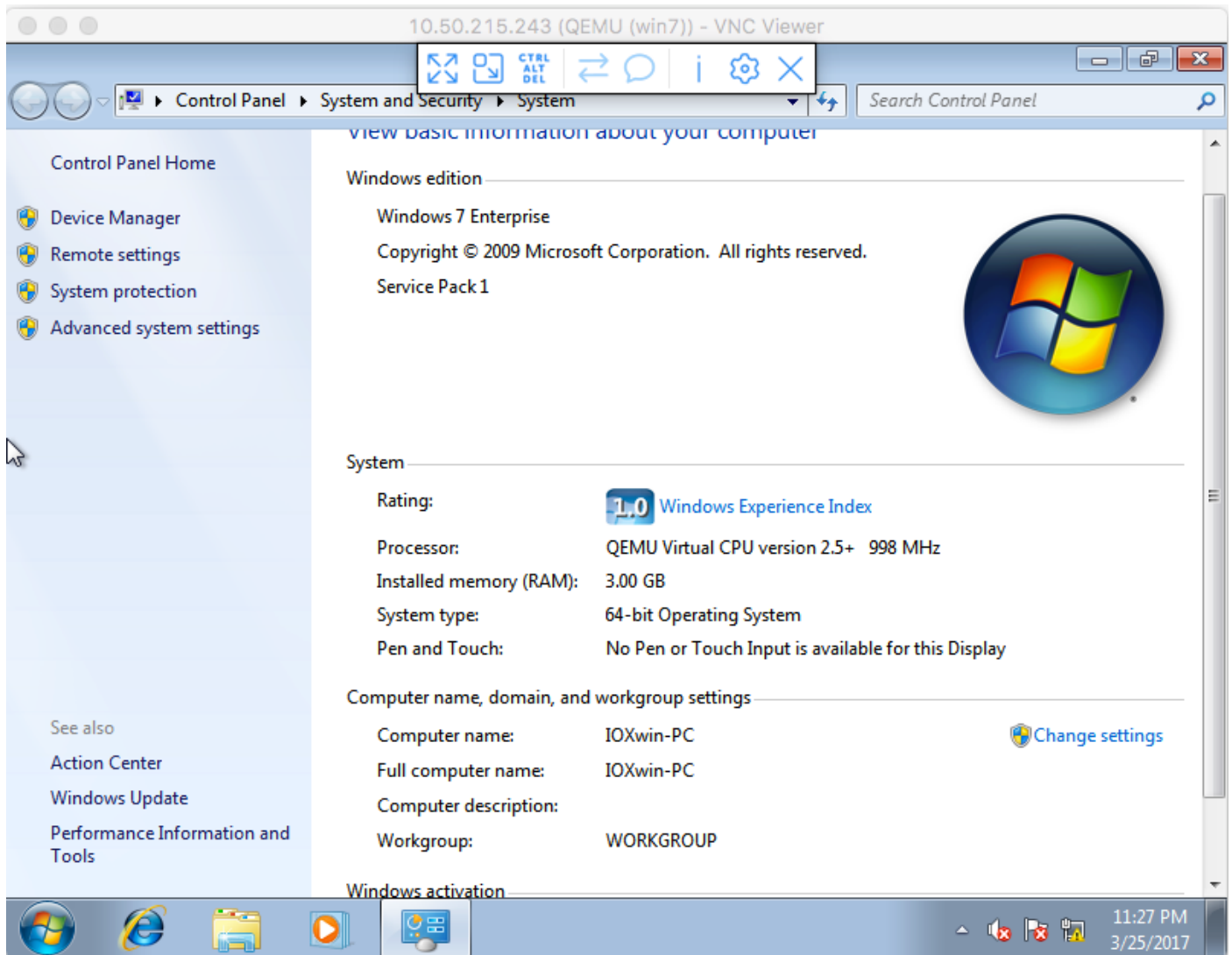
8단계. ioxclient의 마지막 단계는 VM을 시작하는 것입니다.

<#root>

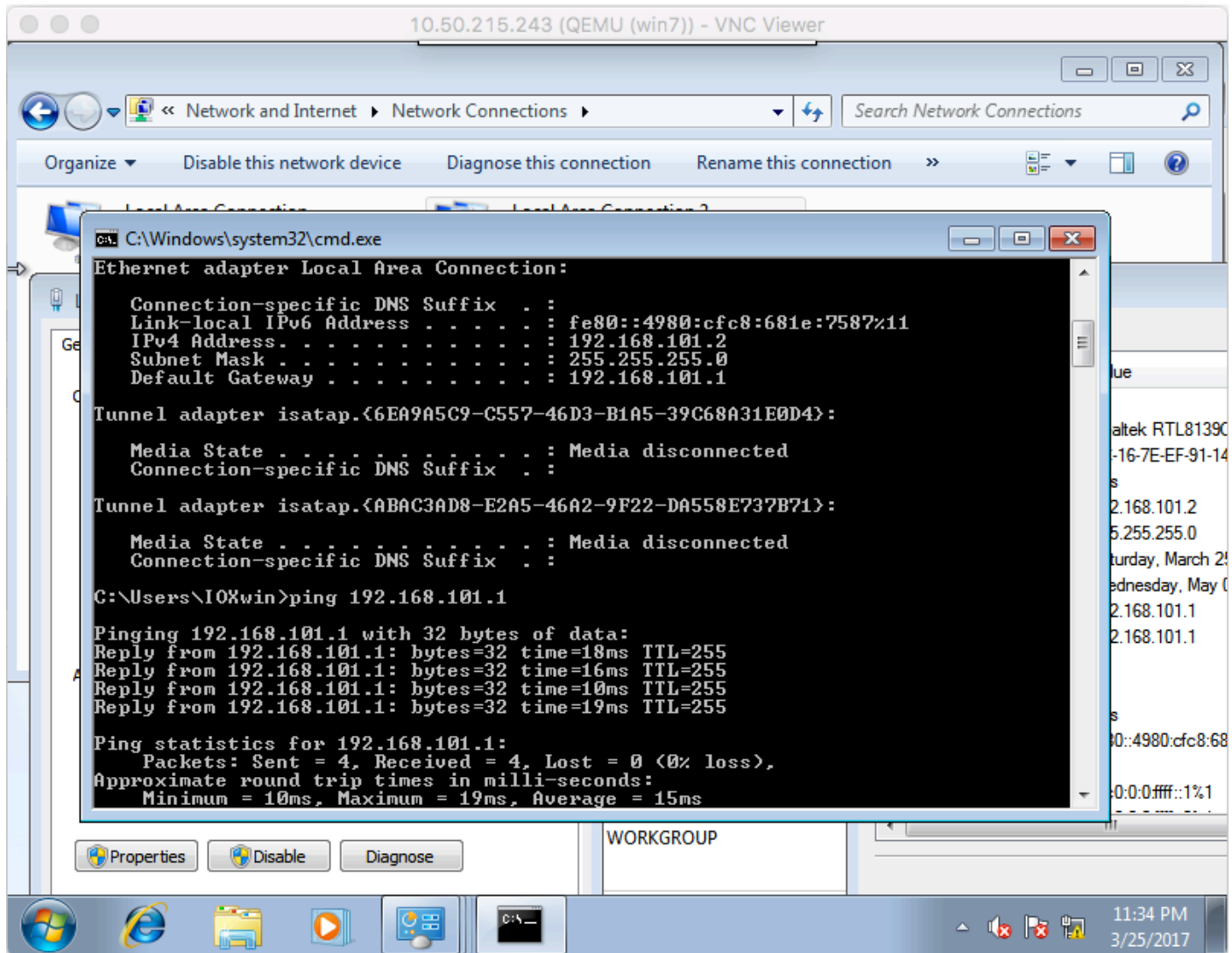
```
[root@cen7 ~]# ./ioxclient app start win7
Currently active profile : default
Command Name:
application-start
App win7 is Started
```

이 시점에서 Windows VM은 CGM-SRV에서 실행되며 사용을 시작할 수 있습니다.

Windows 시스템 콘솔에 액세스하려면 이미지에 표시된 대로 CGR1000 및 포트 5900의 발신 인터페이스에서 VNC 뷰어를 사용할 수 있습니다.



네트워크 관점에서 eth0과 eth1을 win7 IOx VM에 제공하기로 선택했습니다. 그림에 표시된 대로 package.yaml 파일을 사용합니다.



보시다시피 이 인터페이스는 Cisco IOS®에서 실행되는 DHCP 서버에서 IP를 받았으며 추가 컨피그레이션 없이 사용할 수 있습니다.

다음을 확인합니다.

구성이 올바르게 작동하는지 확인하려면 이 섹션을 활용하십시오.

VM이 실행되는지 확인하려면 다음을 수행합니다.

<#root>

```
[root@cen7 ~]# ./ioxclient app list
Currently active profile : CGR1120_20
Command Name:
```

application-list

```
Saving current configuration
List of installed App :
1. win7      --->    RUNNING
```

이미지에 표시된 대로 로컬 관리자에서 상태를 확인할 수도 있습니다.

The screenshot displays the Cisco IOx Local Manager web interface. The top navigation bar includes 'Applications', 'Cartridges', 'System Info', 'System Setting', 'Middleware Service', and a selected 'win7' tab. Below this, a secondary navigation bar shows 'Resources', 'App-Info' (selected), 'App-Config', 'App-DataDir', and 'Logs'. The main content area is divided into three sections: 'Application information', 'Requested Resource', and 'App Access'. The 'Application information' section lists details for the 'win7' application, including its state (RUNNING), author (Jens Depuydt), and description (Windows 7 VM for CSR-SRV). The 'Requested Resource' section shows allocated CPU, memory, disk, and vCPU units. The 'App Access' section provides SSH and VNC access instructions.

Application information	
ID:	win7
State:	RUNNING
Name:	win7
Cartridge 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

문제 해결

이 섹션에서는 설정 문제 해결을 위해 사용할 수 있는 정보를 제공합니다.

구축 관련 문제를 해결하려면 CGM-SRV 호스트 OS에서 ioxclient 또는 /var/log/caf.log의 출력을 확인하십시오.

모든 리소스(Cisco CAF(Application-Hosting Framework), SSH(Secure Shell), VNC)에 액세스하도록 NAT가 올바르게 구성되었는지 확인합니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.