



Red Hat Enterprise Linux 5

Virtualization

A reference guide for virsch, xm, vmm and xend.

위음 2.0

Red Hat Enterprise Linux 5 Virtualization

A reference guide for virsch, xm, vmm and xend.

엠티 2.0

Landmann

rlandmann@redhat.com

법적 공지

Copyright © 2007 Red Hat Inc..

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

초록

이 가이드는 virsh, xm, vmm, xend를 사용하여 Red Hat Enterprise Linux 5에서 게스트 운영 체제를 구성, 생성, 감시하는 데 필요한 정보를 다루고 있습니다. Red Hat Enterprise Linux 가상화 가이드에서 오차를 발견하시거나 가이드를 발전시킬 좋은 제안이 있으시면 언제든지 저희에게 알려주시기 바랍니다! 버그질라(<http://bugzilla.redhat.com/bugzilla/>)에서 Red Hat Enterprise Linux 제품과 Virtualization_Guide에 대한 버그 리포트를 제출해 주시기 바랍니다.

차례

1장. RED HAT VIRTUALIZATION 시스템 구조	4
2장. 운영 체제 지원	6
3장. 하드웨어 지원	7
4장. RED HAT VIRTUALIZATION 시스템 요구사항	8
5장. 시스템 시작	9
6장. GRUB 설정	10
7장. 게스트 도메인 시작	12
8장. 시스템 시작시 도메인 시작/종료	13
9장. 구성 파일	14
10장. CPU 관리	15
11장. 도메인 이전	16
12장. 네트워크 사용을 위한 구성	17
13장. DOMAIN0 보안	18
14장. 저장 공간	19
15장. VIRSH를 사용한 가상 장치 관리	20
15.1. HYPERVISOR 접속	20
15.2. 가상 장치 생성	20
15.3. XML 덤프 설정	20
15.4. 가상 장치 보류	20
15.5. 가상 장치 재시작	20
15.6. 가상 장치 저장	21
15.7. 가상 장치 복구	21
15.8. 가상 장치 종료	21
15.9. 가상 장치 재부팅	21
15.10. 도메인 종료	21
15.11. 도메인 이름을 도메인 ID로 변환	22
15.12. 도메인 ID를 도메인 이름으로 변환	22
15.13. 도메인 이름을 UUID로 변환	22
15.14. 가상 장치 정보 표시	22
15.15. 노드 정보 표시	22
15.16. 가상 장치 표시	22
15.17. 가상 CPU 정보 표시	23
15.18. 가상 CPU 동질성 설정	23
15.19. 가상 CPU 개수 설정	23
15.20. 메모리 할당량 설정	23
15.21. 최대 메모리 설정	24
15.22. MANAGING VIRTUAL NETWORKS	24
16장. XEND를 이용한 가상 장치 관리	26
17장. 가상 장치 관리자를 사용한 가상 장치 관리	28
17.1. 가상 장치 관리자 구조	28

17.2. 오픈 연결 창	28
17.3. 가상 장치 관리자 창	28
17.4. 가상 장치 상세정보 창	29
17.5. 가상 장치 그래픽형식 콘솔	29
17.6. STARTING THE VIRTUAL MACHINE MANAGER	30
17.7. 새로운 가상 장치 생성	31
17.8. 저장된 장치 복구	40
17.9. DISPLAYING VIRTUAL MACHINE DETAILS	42
17.10. CONFIGURING STATUS MONITORING	45
17.11. 도메인 ID 표시	46
17.12. 가상 장치 상태 표시	47
17.13. 가상 CPU 표시	48
17.14. CPU 사용량 표시	49
17.15. 메모리 사용량 표시	50
17.16. MANAGING A VIRTUAL NETWORK	51
17.17. CREATING A VIRTUAL NETWORK	52
18장. RED HAT VIRTUALIZATION 문제해결	61
18.1. 로그 파일 개요 및 위치	61
18.2. 로그 파일 설명	61
18.3. 주요 디렉토리 위치	61
18.4. 문제해결 도구	62
18.5. 로그를 사용한 문제해결	63
18.6. 직렬 콘솔을 사용한 문제해결	64
18.7. 반 가상화 게스트 콘솔 사용	64
18.8. 완전 가상화 게스트 콘솔 사용	65
18.9. LUN 지속성 구현	65
18.10. SELINUX 고려사항	66
18.11. 게스트 디스크 이미지에서 데이터 사용	67
18.12. 일반 문제해결 상황	67
18.13. 루프 장치 오류	68
18.14. 게스트 생성 오류	68
18.15. 직렬 콘솔 오류	69
18.16. 네트워크 브리지 오류	69
18.17. 랩톱 구성	70
18.18. STARTING DOMAINS AUTOMATICALLY DURING SYSTEM BOOT	72
18.19. DOMAIN0 수정	72
18.20. 게스트 구성 파일	73
18.21. 게스트 구성 파일 클론	74
18.22. MAC 주소를 생성하는 스크립트 구성	74
18.23. 가상 장치 실시간 이전 설정	74
18.24. 오류 메시지 해석	75
18.25. 온라인 문제해결 자료	78
19장. 추가 자료	79
19.1. 유용한 웹사이트	79
19.2. 설치된 문서 자료	79
부록 A. 실습 1	80
부록 B. 실습 2	85
부록 C. REVISION HISTORY	87

1장. RED HAT VIRTUALIZATION 시스템 구조

실용적인 Red Hat Virtualization 시스템은 다중 계층으로 이루어졌으며, 권한을 가진 Red Hat Virtualization 구성요소에 의해 구동됩니다. Red Hat Virtualization은 다수의 게스트 운영 체제를 실행할 수 있습니다. 각 게스트 운영 체제는 해당하는 도메인 안에서 실행되며, Red Hat Virtualization은 기존의 실제 CPU 사용을 최적화하기 위해 가상 장치에 가상 CPU를 할당합니다. 각 게스트 운영 체제는 개별적으로 프로그램을 가지며, 또한 각 프로그램을 운영 체제에 맞게 조작할 수 있습니다.

Red Hat Virtualization은 다음 두 가지 중 하나를 선택하여 실행됩니다: **full virtualization(완전 가상화)** 또는 **paravirtualization(반 가상화)**. 완전 가상화는 기본 실제 시스템을 완전히 추출하여 게스트 운영 체제를 실행할 수 있는 새로운 가상 시스템을 생성합니다. 따라서, 게스트 OS와 프로그램을 수정할 필요가 없습니다 (게스트 OS와 프로그램은 가상 환경을 인식하지 못하며, 평소대로 작동합니다). 반 가상화는 가상 장치에서 실행되는 게스트 운영 체제에 사용자 수정이 필요하며(게스트 운영 체제는 가상 장치에서 실행되고 있다는 것을 인식합니다), 거의 완전한 수행 능력을 제공합니다. 사용자는 가상화 구조에 따라 반 가상화와 완전 가상화를 모두 사용할 수도 있습니다.

domain0로 알려진 첫 번째 도메인은 시스템을 시작할 때 자동으로 생성됩니다. **domain0**는 권한을 가진 게스트로서 새로운 도메인을 만들고 이에 대한 가상 장치를 관리할 수 있는 관리 능력을 수행합니다. **domain0**는 또한 네트워크 카드, 하드 디스크 컨트롤러와 같은 실제 하드웨어를 다루며, 다른 가상 장치에 게스트 도메인 보류, 계속, 도입 등과 같은 관리 작업을 수행합니다.

The **hypervisor** (Red Hat's Virtual Machine Monitor) is a virtualization platform that allows multiple operating systems to run on a single host simultaneously within a full virtualization environment. A guest is an operating system (OS) that runs on a virtual machine in addition to the host or main OS.

With Red Hat Virtualization, each guests **memory** comes from a slice of the host's physical memory. For paravirtual guests, you can set both the initial memory and the maximum size of the virtual machine. You can add (or remove) physical memory to the virtual machine at runtime without exceeding the maximum size you specify. This process is called ballooning.

각 게스트를 다수의 가상 **cpu(vcpu**라고 불림)를 사용해 구성할 수 있습니다. Virtual Machine Manager는 실제 CPU의 작업량에 따라서 vcpu를 할당합니다.

게스트에 다수의 **가상 디스크**를 할당할 수 있습니다. 게스트는 가상 디스크를 하드 디스크 또는 CD-ROM 드라이브(완전 가상 게스트)로 인식합니다. 게스트에 호스트의 블록 장치 또는 일반 파일로부터 각 가상 디스크가 제공됩니다. 호스트 장치는 게스트의 파티션 테이블, 다중 파티션 또는 LVM 물리적 볼륨을 포함하는 완전한 디스크 이미지를 가지고 있습니다.

가상 네트워크 인터페이스는 게스트에서 실행됩니다. 다른 인터페이스도 가상 이더넷 인터넷 카드처럼 게스트에서 실행될 수 있습니다. 이러한 네트워크 인터페이스 카드는 영구적인 가상 **MAC(Media Access Control)** 주소를 사용해 구성됩니다. 새로운 게스트 설치 시 디폴트로 **VNIC**를 1천 6백만 주소가 지정된 데이터베이스로부터 임의로 선택된 **MAC** 주소를 할당하여 설치합니다. 따라서, 두 게스트가 같은 **MAC** 주소를 갖는 경우는 발생하지 않습니다. 많은 게스트가 있는 복잡한 사이트에는 **MAC** 주소를 수동으로 할당하여 게스트를 네트워크에서 고유하도록 설정할 수 있습니다.

각 게스트에는 호스트에 연결하는 가상의 **텍스트 콘솔**이 있습니다. 사용자는 게스트 로그인과 콘솔 출력을 텍스트 콘솔로 전환할 수 있습니다.

특정 게스트를 실제 호스트의 일반 비디오 콘솔에 해당하는 가상 **그래픽 콘솔**을 사용하도록 설정할 수 있습니다. 이것은 완전 가상 또는 반 가상 게스트에서 사용 가능합니다. 시작 메시지, 그래픽형식 시작, 다중 가상 터미널 같은 표준 그래픽 아답터 기능을 도입하며 x 윈도우 시스템을 실행할 수 있습니다. 또한, 그래픽형식 키보드를 사용하여 가상 키보드와 마우스를 구성할 수도 있습니다.

Guests can be identified in any of three **identities**: domain name (domain-name), identity (domain-id), or UUID. The domain-name is a text string that corresponds to a guest configuration file. The domain-name is used to launch the guests, and when the guest runs the same name is used to identify and control it. The domain-id is a unique, non-persistent number that gets assigned to an active domain and is used to

identify and control it. The UUID is a persistent, unique identifier that is controlled from the guest's configuration file and ensures that the guest is identified over time by system management tools. It is visible to the guest when it runs. A new UUID is automatically assigned to each guest by the system tools when the guest first installs.

2장. 운영 체제 지원

Red Hat Virtualization's paravirtualization mode allows you to utilize high performance virtualization on architectures that are potentially difficult to virtualize such as x86 based systems. To deploy paravirtualization across your operating system(s), you need access to the paravirtual guest kernels that are available from a respective Red Hat distro (for example, RHEL 4.0, RHEL 5.0, etc.). Whilst your operating system kernels must support Red Hat Virtualization, it is not necessary to modify user applications or libraries.

Red Hat Virtualization은 Intel VT 또는 AMD SVM CPU 하드웨어를 사용할 때 수정되지 않은 게스트 커널을 실행할 수 있도록 해줍니다. 따라서, 사용자가 Intel VT 또는 AMD SVM 시스템에서 이 구조를 사용할 때 운영 체제를 포트화할 필요는 없습니다. Red Hat Virtualization은 다음을 지원합니다:

- 완전 가상화와 반 가상화에 필요한 Intel VT-x 또는 AMD-V Pacifica와 Vanderpool 기술.
- i64의 Intel VT-i.
- NetBSD, FreeBSD, Solaris와 같은 Linux와 UNIX 운영 체제.
- Microsoft Windows as an unmodified guest operating system with Intel Vanderpool or AMD's Pacifica technology.

Hardware-assisted Virtual Machine (HVM), Intel 또는 AMD 플랫폼을 사용하는 시스템에 완전 가상 게스트를 실행하려면, CPU가 이러한 기능을 수행하는 데 적합한 능력을 갖추고 있는지 반드시 확인해야 합니다.

Intel 지원에 필요한 CPU 플래그가 있는지 확인하려면, 다음을 실행하시기 바랍니다:

```
grep vmx /proc/cpuinfo
```

다음과 같이 출력됩니다:

```
flags      : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm syscall nx lm constant_tsc pni monitor
ds_cpl vmx est tm2 cx16 xtpr lahf_lm
```

vmx 플래그가 표시되면, 시스템은 Intel 지원을 갖추고 있습니다.

AMD 지원에 필요한 CPU 플래그가 있는지 확인하려면, 다음을 실행하시기 바랍니다:

```
grep svm /proc/cpuinfo
cat /proc/cpuinfo | grep svm
```

다음과 같이 출력됩니다:

```
flags      : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dt
acpi mmx fxsr sse sse2 ss ht tm syscall nx mmtxt fxsr_opt rdtscp lm
3dnowext pni cx16 lahf_lm cmp_legacy svm cr8_legacy
```

svm 플래그가 표시되면, 시스템은 AMD 지원을 갖추고 있습니다.



참고

CPU 플래그를 확인하는 것 외에 시스템의 BIOS에서 완전 가상화를 활성화하셔야 합니다.

3장. 하드웨어 지원

Red Hat Virtualization은 멀티프로세서 시스템을 지원하며, 다음과 같은 P6 클래스(또는 이전 클래스) 프로세서를 사용하는 x86 구조 시스템에서 Red Hat Virtualization을 실행할 수 있습니다.

- Celeron
- Pentium II
- Pentium III
- Pentium IV
- Xeon
- AMD Athlon
- AMD Duron

Red Hat Virtualization에서 32 비트 호스트는 단지 32 비트 반 가상 게스트를 실행하고, 64 비트 호스트는 64 비트 반 가상 게스트만을 실행합니다. 또한, 64 비트 완전 가상 호스트는 32 비트, 32 비트 PAE, 64 비트 게스트를 실행하고, 32 비트 완전 가상 호스트는 PAE 완전 가상 게스트와 PAE가 아닌 완전 가상 게스트를 모두 실행합니다.

Red Hat Enterprise Linux Virtualization 커널은 x86_64 시스템에서 32GB 이상의 메모리를 지원하지 않습니다. 32GB 이상의 실제 메모리가 설치된 시스템에서 가상화 커널을 시작하려면, 커널 명령행에 **mem=32G**를 추가해야 합니다. 아래 예제는 **grub.conf** 파일에 어떻게 적절한 파라미터를 활성화하는지 보여줍니다.

```
title Red Hat Enterprise Linux Server (2.6.18-4.elxen)
root (hd0, 0)
kernel /xen.gz-2.6.18-4-el5 mem=32G
module /vmlinuz -2.6.18-4.el5xen ro root=LABEL=/
module /initrd-2.6.18-4.el5xen.img
```

PAE(Physical Address Extension)은 사용자 프로그램에 필요한 가상 또는 실제 메모리의 양을 증가시켜 주는 기술입니다. Red Hat Virtualization을 사용하려면 시스템에 PAE를 활성화해야 합니다. PAE 지원을 사용한 Red Hat Virtualization 32 bit 구조는 최대 16GB의 실제 메모리를 지원하며, 시스템에서 실행되는 모든 게스트에 최소 256 메가바이트 이상의 RAM을 할당하기를 권장합니다. Red Hat Virtualization은 x86/64 장치를 최대 64GB 메모리까지 사용하도록 활성화할 수 있습니다. Red Hat Virtualization 커널은 PAE를 지원하지 않는 시스템에서는 실행할 수 없습니다. 사용자의 시스템이 PAE를 지원하는지 확인하기 위해서는 다음 명령어를 입력하시기 바랍니다:

```
grep pae /proc/cpuinfo
```

다음과 같이 출력됩니다:

```
flags : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 mmx fxsr sse
syscall mmtxt 3dnowext 3dnow up ts
```

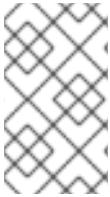
출력이 위와 일치하면 (또는 비슷하면) CPU는 PAE 지원을 갖추고 있으며, 아무런 출력이 없으면 CPU는 PAE 지원을 갖추고 있지 않는 것입니다.

4장. RED HAT VIRTUALIZATION 시스템 요구사항

Red Hat Virtualization 시스템에서는 다음 아이템이 요구됩니다:

- 사용 가능한 Red Hat RHEL 5 Linux 배포 버전
- 사용 가능한 GRUB 부트로더
- Root 사용 권한
- P6 클래스(또는 이전 클래스) 프로세서
- Linux bridge-utils
- Linux 핫플러그 시스템
- zlib 개발 설치
- Python 2.2 런타임
- initscript

종속성은 설치 과정 중에 자동으로 설정됩니다.



참고

If your system CPU architecture is ia64, you need to manually install the **xen-ia64-guest-firmware** package to run a fully virtualized guest. This package is provided in the Supplementary CD and is not installed by default.

5장. 시스템 시작

Red Hat Virtualization 구성요소를 설치하신 후에, 시스템을 반드시 재부팅하셔야 합니다. 시작하였으면 평소처럼 시스템에 로그인하시기 바랍니다. Red Hat Virtualization을 시작하기 전에 반드시 루트로 로그인하셔야 합니다. `initscript`가 `xend` 제어 데몬을 이미 시작하지만 다음을 입력하여 `xend`를 수동으로 시작할 수 있습니다:

```
service xend start
```

시작 시 `xend`를 활성화하려면, `chkconfig xend` 명령어를 사용하시기 바랍니다.

`xend` 노드 제어 데몬은 가상 장치와 관련된 시스템 관리 기능을 수행합니다. 이 데몬은 가상화된 자원을 제어하며, `xend`는 가상 장치와 연동되도록 반드시 실행되어야 합니다. `xend`를 시작하기 전에, `etc/xen` 디렉토리에 있는 `xend-config.sxp` 라는 `xend` 구성 파일을 편집하여 운영 파라미터를 지정해야 합니다.

6장. GRUB 설정

GNU Grand Unified Boot Loader (or GRUB) is a program which enables the user to select which installed operating system or kernel to load at system boot time. It also allows the user to pass arguments to the kernel. The GRUB configuration file (located in `/boot/grub/grub.conf`) is used to create a list of operating systems to boot in GRUB's menu interface. When you install the kernel-xen RPM, a post script adds kernel-xen entries to the GRUB configuration file. You can edit the `grub.conf` file and enable the following GRUB parameter:

```
title Red Hat Enterprise Linux Server (2.6.18-3.el5xen)
root    (hd0; 0)
kernel  /xen.gz.-2.6.18-3.el5
module  /vmlinuz-2.6..18-3.el5xen ro root=/dev/VolGroup00/LogVol100 rhgb
quiet
module  /initrd-2.6.18-3. el5xenxen.img
```

리눅스 grub 항목을 위 예제와 같이 설정하면, 부트로더는 hypervisor(`initrd` 이미지)와 리눅스 커널을 불러들입니다. 이 커널 항목이 다른 항목보다 위에 있으므로 이 커널이 먼저 메모리에 상주하게 됩니다. 부트로더는 hypervisor와 Linux 커널로 또는 그것들로부터 명령행 인수를 보내거나 받습니다. 다음 예제 항목은 어떻게 domain0 리눅스 커널 메모리를 800MB로 제한하는지를 보여줍니다.

```
title Red Hat Enterprise Linux Server (2.6.18-3.el5xen)
root    (hd0; 0)
kernel  /xen.gz.-2.6.18-3.el5 dom0_mem=800M
module  /vmlinuz-2.6..18-3.el5xen ro root=/dev/VolGroup00/LogVol100 rhgb
quiet
module  /initrd-2.6.18-3. el5xenxen.img
```

다음의 GRUB 파라미터를 사용하여 Virtualization hypervisor를 구성할 수 있습니다:

```
mem
```

domain0에 필요한 메모리의 양을 제한합니다.

```
com1=115200, 8n1
```

시스템의 첫째 시리얼 포트를 시리얼 콘솔로 작동하도록 활성화합니다 (com2는 다음 포트로 할당, 기타 등등..).

```
dom0_mem
```

domain0에 필요한 메모리의 양을 제한합니다.

```
dom0_max_vcpus
```

domain0에 보이는 CPU 수를 제한합니다.

acpi

ACPI hypervisor를 hypervisor와 domain0으로 전환합니다. ACPI 파라미터 옵션은 다음을 포함합니다:

```
/* **** Linux config options: propagated to domain0 ****/  
/* "acpi=off":      Disables both ACPI table parsing and interpreter.  
*/  
/* "acpi=force":   Overrides the disable blacklist.  
*/  
/* "acpi=strict":  Disables out-of-spec workarounds.  
*/  
/* "acpi=ht":      Limits ACPI from boot-time to enable HT.  
*/  
/* "acpi=noirq":   Disables ACPI interrupt routing.  
*/
```

noacpi

인터럽트 전달을 위해 ACPI를 비활성화시킵니다.

7장. 게스트 도메인 시작

xm 프로그램을 사용하여 게스트 도메인을 시작할 수 있습니다. 또한, **virsh**를 사용하여 Virtual Machine Manager가 게스트 도메인을 시작하도록 설정할 수 있습니다. 게스트 도메인을 시작하기 전에 게스트 호스트를 먼저 설치해야 합니다. 다음 예제는 **xm create** 하위 명령어를 사용합니다:

```
# xm create -c guestdomain1
```

guestdomain1은 사용자가 시작하는 도메인에 필요한 구성 파일입니다. **-c** 옵션을 사용하면 시작 후에 실제 콘솔로 연결합니다.

8장. 시스템 시작시 도메인 시작/종료

어느 때나 도메인을 시작하고 종료할 수 있습니다. `domain0`는 재시작 전에 실행 중인 모든 도메인을 종료하기 위해 대기합니다. 반드시 종료하고자 하는 도메인의 구성 파일을 `/etc/xen/` 디렉토리로 옮기셔야 합니다. 또한, 시스템 시작 시 시작하기를 원하는 모든 도메인은 반드시 `/etc/xen/auto`로 심볼릭 링크 되어야 합니다.

```
chkconfig xendomains on
```

chkconfig xendomains on 명령어는 자동으로 도메인을 시작하지 않으며, 다음 시스템 시작 때 적용됩니다.

```
chkconfig xendomains off
```

chkconfig xendomains off 명령어는 실행 중인 모든 Red Hat Virtualization 도메인을 종료하며, 다음 시스템 시작 시에 적용됩니다.

9장. 구성 파일

Red Hat Virtualization configuration files contain the following standard variables. Configuration items within these files must be enclosed in quotes ("). These configuration files reside in the `/etc/xen` directory.

표 9.1. Red Hat Virtualization 구성 파일

아이템	설명
pae	physical address extention 구성 데이터 지정.
apic	advanced programmable interrupt controller 구성 데이터 지정.
memory	메가바이트 단위로 메모리 크기 지정.
vcpus	가상 CPU 개수 지정.
console	도메인 콘솔로 익스포트하는 포트 번호 지정.
nic	가상 네트워크 인터페이스 번호 지정.
vif	Lists the randomly-assigned MAC addresses and bridges assigned to use for the domain's network addresses.
disk	도메인으로 익스포트할 블록 장치와 읽기 전용 권한을 사용하여 실제 장치를 도메인에 익스포트하는 블록 장치 목록.
dhcp	DHCP를 사용하여 네트워크 활성화.
netmask	구성된 IP 넷마스크 지정.
gateway	구성된 IP 게이트웨이 지정.
acpi	advanced configuration power interface 구성 데이터 지정.

10장. CPU 관리

Red Hat Virtualization allows a domain's virtual CPUs to associate with one or more host CPUs. This can be used to allocate real resources among one or more guests. This approach allows Red Hat Virtualization to make optimal use of processor resources when employing dual-core, hyperthreading, or other advanced CPU technologies. If you are running I/O intensive tasks, its typically better to dedicate either a hyperthread or entire core to run domain0. The Red Hat Virtualization credit scheduler automatically rebalances virtual cpus between physical ones, to maximize system use. The Red Hat Virtualization system allows the credit scheduler to move CPUs around as necessary, as long as the virtual CPU is pinned to a physical CPU.

11장. 도메인 이전

도메인 이전은 실행 중인 가상 도메인을 하나의 실제 호스트로부터 다른 호스트로의 이동을 말합니다. Red Hat Virtualization은 다음 두 가지의 이전 방식을 지원합니다 — 오프라인과 실시간. 오프라인 이전은 가상 장치를 호스트에서 다른 호스트로 이동시키는 동안 가상 장치를 정지시키고, 메모리를 전송하고, 목적지 호스트에서 다시 작동시킵니다. 실시간 이전은 같은 작업을 하지만, 도메인에 직접적인 영향을 주지는 않습니다. 실시간 이전을 수행하면, 도메인은 평소 작업을 계속하며, 사용자 관점에서 이를 알아채지 못합니다. 실시간 이전을 실행하려면, 두 호스트에는 모두 Red Hat Virtualization과 xend 데몬이 실행 중이어야 합니다. 또한, 목적지 호스트는 이전 후에 도메인 대역폭을 수용하기 위한 충분한 자원(예, 메모리 용량)을 가지고 있어야 합니다. 출발지와 목적지 장치는 반드시 같은 구조와 가상화 확장(예, i386-VT, x86-64-VT, x86-64-SVM 등)을 갖추고 있으며, 같은 L2 서브넷상에 위치하고 있어야 합니다.

When a domain migrates its MAC and IP addresses move with it. Only virtual machines with the same layer-2 network and subnets will successfully migrate. If the destination node is on a different subnet, the administrator must manually configure a suitable EtherIP or IP tunnel in the remote node of domain0. The xend daemon stops the domain and copies the job over to the new node and restarts it. The Red Hat Virtualization RPM does not enable migration from any other host except the localhost (see the `/etc/xend-config.sxp` file for information). To allow the migration target to accept incoming migration requests from remote hosts, you must modify the target's `xen-relocation-hosts-allow` parameter. Be sure to carefully restrict which hosts are allowed to migrate, since there is no authentication.

도메인 이전 시 많은 파일을 옮겨야 하기에, 이 작업은 많은 시간이 소모됩니다. 도메인을 오픈 네트워크 연결을 사용하여 이전하려면, 목적지 호스트 연결을 유지해야 하며 SSH 연결 또한 작동해야 합니다. Red Hat Virtualization의 기본 iptable 규칙에 따르면, 들어오는 도메인 이전 연결은 허용되지 않습니다. 따라서, 특정 iptable 규칙을 생성하여 연결을 허용해야 합니다.

다음 `xm migrate` 명령어를 사용하여 오프라인 도메인 이전을 수행합니다:

```
xm migrate domain-id [destination domain]
```

다음 `xm migrate` 명령어를 사용하여 실시간 도메인 이전을 수행합니다:

```
xm migrate domain-id -l [destination domain]
```

You may need to reconnect to the domain's console on the new machine. You can use the `xm console` command to reconnect.

12장. 네트워크 사용을 위한 구성

Red Hat Virtualization을 사용자 네트워크 구조로 통합하는 과정은 매우 복잡하며, 시스템 구조에 따라서 다중 이더넷 인터페이스 사용과 브리징 설정을 사용자 환경에 맞추어 구성해야 할 것입니다.

Each domain network interface is connected to a virtual network interface in **dom0** by a point to point link. These devices are **vif <domid>** and **<vifid>**. **vif1.0** for the first interface in domain 1; **vif3.1** for the second interface in domain 3.

domain0는 브리징, 라우팅, 비율 제한 등의 표준 리눅스 규약을 사용하여 가상 인터페이스의 네트워크의 사소통을 제어합니다. **xend** 데몬은 두 가지의 셸 스크립트를 도입하여 사용자의 네트워크와 새로운 가상 인터페이스의 초기 구성을 수행합니다. 이 스크립트는 모든 가상 인터페이스에 대한 하나의 브리지를 설정합니다. 스크립트를 사용자 환경에 맞게 수정하여 추가적인 라우팅과 브리징을 설정할 수 있습니다.

Red Hat Virtualization's virtual networking is controlled by the two shell scripts, **network-bridge** and **vif-bridge**. **xend** calls these scripts when certain events occur. Arguments can be passed to the scripts to provide additional contextual information. These scripts are located in the **/etc/xen/scripts** directory. You can change script properties by modifying the **xend-config.sxp** configuration file located in the **/etc/xen** directory.

network-bridge — **xend**가 시작되거나 종료될 때, 이 스크립트는 가상 네트워크를 시작하거나 종료합니다. 구성 시작 시 브리지 **xen-br0**를 생성하고 **eth0**를 브리지로 이동시키며, 이에 맞게 라우팅을 수정합니다. **xend**가 종료되면, 생성된 브리지를 삭제하고 **eth0**를 제거하여 기존 IP와 라우팅 구성을 복구합니다.

vif-bridge - 이 스크립트는 도메인의 모든 가상 인터페이스에 쓰이며, 방화벽 규칙을 설정하며 해당하는 브리지에 **vif**를 추가할 수 있습니다.

Red Hat Virtualization을 구성하는 데 도움이 되는 다른 스크립트가 있습니다: **network-route**, **network-nat**, **vif-route**, **vif-nat**. 이러한 스크립트는 사용자 환경에 맞춘 형태로 대체될 수도 있습니다.

13장. DOMAIN0 보안

When deploying Red Hat Virtualization on your corporate infrastructure, you must ensure that domain0 cannot be compromised. Domain0 is the privileged domain that handles system management. If domain0 is insecure, all other domains in the system are vulnerable. There are several ways to implement security you should know about when integrating Red Hat Virtualization into your systems. Together with other people in your organization, you should create a 'deployment plan' that contains the operating specifications and services that will run on Red Hat Virtualization, and what is needed to support these services. Here are some security issues to consider when putting together a deployment plan:

- 필요한 서비스를 최소화하여 실행하시기 바랍니다. domain0에 너무 많은 작업과 서비스가 있으면 안됩니다. domain0에 실행되는 서비스가 적을수록, 높은 수준의 보안이 제공됩니다.
- domain0을 보안하기 위해 SELinux를 활성화하시기 바랍니다.
- 방화벽을 사용하여 domain0에 대한 의사소통을 제한하시기 바랍니다. default-reject 규칙을 사용하여 방화벽을 설정하면 domain0 공격에 대한 보안을 강화할 수 있습니다. 서비스에 직면한 네트워크를 제한하는 것 또한 중요합니다.
- domain0에 일반 사용자가 접속하지 못하도록 설정하시기 바랍니다. 일반 사용자가 domain0를 사용하도록 허용하면, domain0를 위협에 노출시키는 위협을 자초하는 것입니다. domain0는 권한을 가진 도메인이며, 비권한 계정 허용은 이러한 보안 수준을 위협할 수 있다는 것을 기억해 두시기 바랍니다.

14장. 저장 공간

가상 장치의 저장 공간을 관리하는 데에는 여러 가지 방법이 있습니다. `domain0`의 실제 블록 장치(하드 드라이브 또는 파티션)를 가상 블록 장치(VBD)로서 게스트 도메인에 익스포트할 수 있습니다. 또한, 파일 기반 VBD로서 파티션된 이미지로부터 직접 익스포트할 수도 있습니다. Red Hat Virtualization은 설치 중에 디폴트로 LVM과 `blktap`를 활성화합니다. 또한, NFS, CLVM, iSCSI 같은 표준 네트워크 프로토콜을 도입하여 가상 장치에 필요한 저장 공간을 제공할 수 있습니다.

15장. VIRSH를 사용한 가상 장치 관리

virsh 프로그램을 사용하여 가상 장치를 관리할 수 있습니다. 이 유틸리티는 **libvirt** 관리 API로 만들어졌으며, **xm** 또는 그래픽형식 **Virtual Machine Manager**를 대체하는 도구로서 사용됩니다. 비권한 사용자는 이 유틸리티를 사용하여 읽기 전용 운영을 수행할 수 있습니다. **xend/qemu** 명령어를 실행하려면, **xend/qemu**를 서비스로 작동하도록 활성화해야 합니다. 구성 파일을 수정한 후에 시스템을 재부팅하면 **xend/qemu**는 서비스로 작동할 것입니다. **virsh**를 사용하여 **vm** 작업을 스크립트할 수 있습니다. 또한, **xm** 도구와 같이 명령행에서 **virsh**를 실행할 수 있습니다.

15.1. HYPERVISOR 접속

virsh를 사용하여 hypervisor 세션을 시작합니다:

```
virsh connect <name>
```

Where **<name>** is the machine name of the hypervisor. If you want to initiate a read-only connection, append the above command with **-readonly**.

15.2. 가상 장치 생성

XML 장치 정의로부터 새로운 가상 장치 세션을 만들 수 있습니다. 이전에 **xm** 도구를 사용하여 이미 만들어진 게스트에 대한 가상 장치 또한 생성할 수 있습니다.

```
virsh create <path to XML configuration file>
```

15.3. XML 덤프 설정

virsh를 사용하여 존재하는 가상 장치에 대한 데이터 덤프를 수행할 수 있습니다.

```
virsh dumpxml [domain-id | domain-name | domain-uuid]
```

This command outputs the domain information (in XML) to **stdout** . If you save the data to a file, you can use the **create** option to recreate the virtual machine.

15.4. 가상 장치 보류

virsh를 사용하여 도메인을 보류시킬 수 있습니다:

```
virsh suspend [domain-id | domain-name | domain-uuid]
```

When a domain is in a suspended state, it still consumes system RAM. There will also be no disk or network I/O when suspended. This operation is immediate and the virtual machine must be restarted with the **resume** option.

15.5. 가상 장치 재시작

virsh를 사용하여 보류 중인 가상 장치를 복구할 수 있습니다:

```
virsh resume [domain-id | domain-name | domain-uuid]
```


이 기능은 일시적이며, 가상 장치 파라미터는 **suspend**와 **resume** 주기에서 유지됩니다.

15.6. 가상 장치 저장

virsh를 사용하여 가상 장치의 현재 상태를 파일에 저장할 수 있습니다:

```
virsh save [domain-name][domain-id | domain-uuid][filename]
```

This stops the virtual machine you specify and saves the data to a file, which may take some time given the amount of memory in use by your virtual machine. You can restore the state of the virtual machine with the **restore** option.

15.7. 가상 장치 복구

You can use **virsh** to restore a virtual machine that you previously saved with the **virsh save** option:

```
virsh restore [filename]
```

This restarts the saved virtual machine, which may take some time. The virtual machine's name and UUID are preserved but are allocated for a new id.

15.8. 가상 장치 종료

virsh를 사용하여 가상 장치를 종료할 수 있습니다:

```
virsh shutdown [domain-id | domain-name | domain-uuid]
```

xmdomain.cfg 파일에 **on_shutdown** 파라미터를 수정하여 가상 장치 종료에 대한 행동을 제어할 수 있습니다.

15.9. 가상 장치 재부팅

virsh를 사용하여 가상 장치를 재부팅할 수 있습니다:

```
virsh reboot [domain-id | domain-name | domain-uuid]
```

xmdomain.cfg 파일에 **on_reboot** 파라미터를 수정하여 가상 장치 재부팅에 대한 행동을 제어할 수 있습니다.

15.10. 도메인 종료

virsh를 사용하여 가상 장치를 종료할 수 있습니다:

```
virsh destroy [domain-name | domain-id | domain-uuid]
```

This command does an immediate ungraceful shutdown and stops any guest domain sessions (which could potentially lead to file corrupted filesystems still in use by the virtual machine). You should use the **destroy** option only when the virtual machine's operating system is non-responsive. For a paravirtualized virtual machine, you should use the **shutdown** option.

15.11. 도메인 이름을 도메인 ID로 변환

virsh를 사용하여 도메인 이름 또는 UUID를 도메인 id로 변환할 수 있습니다:

```
virsh domid [domain-name | domain-uuid]
```

15.12. 도메인 ID를 도메인 이름으로 변환

virsh를 사용하여 도메인 id 또는 UUID를 도메인 이름으로 변환할 수 있습니다:

```
virsh domname [domain-name | domain-uuid]
```

15.13. 도메인 이름을 UUID로 변환

virsh를 사용하여 도메인 이름을 UUID로 변환할 수 있습니다:

```
virsh domuuid [domain-id | domain-uuid]
```

15.14. 가상 장치 정보 표시

virsh를 사용하여 도메인 ID로 구분된 주어진 가상 장치에 대한 정보를 표시할 수 있습니다:

```
virsh dominfo [domain-id | domain-name | domain-uuid]
```

15.15. 노드 정보 표시

virsh를 사용하여 노드 정보를 표시할 수 있습니다:

```
virsh nodeinfo
```

출력은 다음과 비슷합니다:

```
CPU model           x86_64
CPU (s)             8
CPU frequency       2895 Mhz
CPU socket(s)       2
Core(s) per socket  2
Threads per core:   2
Numa cell(s)        1
Memory size:        1046528 kb
```

가상화 프로세스를 지원하는 노드 정보와 장치를 표시합니다.

15.16. 가상 장치 표시

virsh를 사용하여 가상 장치 목록과 현재 상태를 표시할 수 있습니다:

```
virsh list domain-name [ --inactive | -- -all]
```

—**inactive** 옵션은 활성화되지 않은 도메인(현재 활성화되지 않도록 정의된 도메인) 목록을 표시합니다. — **-all** 옵션은 활성화 상태와 관계없이 모든 도메인 목록을 표시합니다. 출력 내용은 다음과 비슷합니다:

ID	Name	State
0	Domain0	running
1	Domain202	paused
2	Domain010	inactive
3	Domain9600	crashed

다음 여섯 가지 상태가 있습니다:

```
running lists domains currently active on the CPU
blocked lists domains that are blocked
paused lists domains that are suspended
shutdown lists domains that are in process of shutting down
shutoff lists domains that are completely down.
crashed lists domains that are crashed
```

15.17. 가상 CPU 정보 표시

virsh 를 사용하여 가상 장치로부터 가상 CPU 정보를 표시합니다.

```
virsh vcpuinfo [domain-id | domain-name | domain-uuid]
```

15.18. 가상 CPU 동질성 설정

virsh를 사용하여 실제 CPU에서 가상 CPU의 동질성을 설정할 수 있습니다:

```
virsh vcpupin [domain-id | domain-name | domain-uuid] [vcpu] , [cpulist]
```

여기서 **[vcpu]**는 가상 VCPU 번호이며 **[cpulist]**는 CPU의 실제 번호를 표시합니다.

15.19. 가상 CPU 개수 설정

You can use **virsh** to modify a Virtual Machine's number of CPUs:

```
virsh setvcpus [domain-name | domain-id | domain-uuid] [count]
```

새로운 값은 가상 장치를 생성할 때 지정했던 개수를 초과할 수 없음을 주의하시기 바랍니다.

15.20. 메모리 할당량 설정

You can use **virsh** to modify a domain's memory allocation:

```
virsh setmem [domain-id | domain-name] [count]
```

You must specify the **[count]** in kilobytes. Note that the new count cannot exceed the amount you specified when you created the Virtual Machine. Values lower than 64 MB probably won't work. You can adjust the Virtual Machine memory as necessary.

15.21. 최대 메모리 설정

You can use **virsh** to modify a Virtual Machine's maximum memory:

```
virsh setmaxmem [domain-name | domain-id | domain-uuid] [count]
```

You must specify the [count] in kilobytes. Note that the new count cannot exceed the amount you specified when you created the Virtual Machine. Values lower than 64 MB probably won't work. The maximum memory doesn't affect the current use of the Virtual Machine (unless the new value is lower which should shrink memory usage).

15.22. MANAGING VIRTUAL NETWORKS

You can use **virsh** to manage virtual networks. To list virtual networks:

```
virsh net-list
```

This command generates output similar to:

```
[root@domain ~]# virsh net-list
Name                State      Autostart
-----
default             active    yes
vnet1               active    yes
vnet2               active    yes
```

To view network information for a specific virtual network:

```
virsh net-dumpxml [vnet name]
```

This displays information about a specified virtual network in XML format:

```
[root@domain ~]# virsh net-dumpxml vnet1
<network>
  <name>vnet1</name>
  <uuid>98361b46-1581-acb7-1643-85a412626e70</uuid>
  <forward dev='eth0' />
  <bridge name='vnet0' stp='on' forwardDelay='0' />
  <ip address='192.168.100.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.100.128' end='192.168.100.254' />
    </dhcp>
  </ip>
</network>
```

Other **virsh** commands used in managing virtual networks are:

- **virsh net-autostart** [*network name*] — Autostart a network specified as [network name]
- **virsh net-create** [*XML file*] — Generates and starts a new network using a preexisting XML file
- **virsh net-define** [*XML file*] — Generates a new network from a preexisting XML file without starting it
- **virsh net-destroy** [*network name*] — Destroy a network specified as [network name]
- **virsh net-name** [*network UUID*] — Convert a specified [network UUID] to a network name
- **virsh net-uuid** [*network name*] — Convert a specified [network name] to a network UUID
- **virsh net-start** [*name of an inactive network*] — Starts a previously undefined inactive network
- **virsh net-undefine** [*name of an inactive network*] — Undefine an inactive network

16장. XEND를 이용한 가상 장치 관리

xend 노드 제어 데몬은 가상 장치와 관련된 특정 시스템 관리 기능을 수행합니다. 이 데몬은 가상 자원을 제어하며, **xend**은 반드시 가상 장치와 연동하도록 실행되어야 합니다. **xend**을 시작하기 전에, 반드시 **etc/xen** 디렉토리에 있는 **xend-config.sxp**라는 **xend** 구성 파일을 편집하여 운영 파라미터를 지정해야 합니다. 다음은 **xend-config.sxp** 구성 파일에서 활성화 또는 비활성화할 수 있는 파라미터를 보여줍니다:

표 16.1. Red Hat Virtualization xend 구성 파라미터

아이템	설명
console-limit	Determines the console server's memory buffer limit and assigns values on a per-domain basis
min-mem	domain0에 사용되는 메가바이트의 최소 수를 결정 (0을 입력하면, 값은 변하지 않음).
dom0 cpus	domain0에서 사용되는 CPU 개수를 결정 (디폴트로 최소 1 CPU 할당).
enable-dump	충돌 발생 시 덤프 활성화 여부 결정 (디폴트: 0).
external-migration-tool	외부 장치 이전을 제어하는 스크립트와 프로그램을 결정 (etc/xen/scripts/external-device-migrate 에 스크립트 위치).
logfile	로그 파일의 위치 결정 (디폴트: /var/log/xend.log).
loglevel	로그 모드 값 선별: DEBUG, INFO, WARNING, ERROR, CRITICAL (디폴트: DEBUG).
network-script	네트워크 환경을 활성화하는 스크립트 결정 (스크립트는 반드시 etc/xen/scripts 디렉토리에 위치).
xend-http-server	http 스트림 관리 서버 활성화 (디폴트: no).
xend-unix-server	유닉스 도메인 소켓 서버 활성화 (소켓 서버는 의사소통 중단점으로서 낮은 수준 네트워크 연결을 제어하고 들어오는 연결을 허용하거나 거절).
xend-relocation-server	장치 교체 이전을 위한 서버 재배포 활성화 (디폴트: no).
xend-unix-path	xend-unix-server 명령어 출력 데이터가 있는 위치 결정 (디폴트: var/lib/xend/xend-socket).
xend-port	http 관리 서버가 사용하는 포트 결정 (디폴트: 8000).
xend-relocation-port	재배포 서버가 사용하는 포트 결정 (디폴트: 8002).

아이템	설명
xend-relocation-address	시스템 이전에 사용되는 가상 장치 주소 결정.
xend-address	도메인 소켓 서버가 바인드하는 주소 결정.

이러한 운영 파라미터를 설정한 후에, **xend**가 실행되고 있는지 확인해야 하며, 실행되고 있지 않으면 데몬을 시작해야 합니다. 명령어 프롬프트에서 다음 명령어를 입력하여 **xend** 데몬을 시작하시기 바랍니다:

```
service xend start
```

다음을 사용하여 **xend** 데몬을 정지시킬 수 있습니다:

```
service xend stop
```

위 명령어는 실행 중인 데몬을 정지시킵니다.

다음을 사용하여 **xend** 데몬을 재시작할 수 있습니다:

```
service xend restart
```

데몬은 다시 시작합니다.

xend 데몬의 상태를 확인할 수 있습니다.

```
service xend status
```

The output displays the daemon's status.

17장. 가상 장치 관리자를 사용한 가상 장치 관리

이 단원은 Red Hat Virtualization 가상 장치 관리자(VMM)의 창, 대화상자, 다양한 GUI 제어에 대해 설명합니다.

17.1. 가상 장치 관리자 구조

Red Hat Virtualization은 호스트에 사용되고 가상 장치를 관리하는 소프트웨어 구성요소 모음입니다. 가상 장치 관리자(VMM)는 사용자 시스템의 가상 장치를 그래픽 형식으로 보여주며, 반 가상 장치와 완전 가상 장치를 모두 정의할 수 있습니다. 가상 장치 관리자를 사용하여 메모리 할당, 가상 CPU 할당, 운영 수행 감시, 가상 시스템 저장, 복구, 정지, 재시작, 종료와 같은 모든 가상화 관리 작업을 수행할 수 있습니다. 또한, VMM을 사용하여 텍스트형식과 그래픽형식 콘솔을 사용할 수 있습니다. Red Hat Virtualization은 기초 하드웨어와 네트워크 구성으로부터 CPU와 메모리 자원을 가져옵니다. 따라서, 자원을 영역화시키고 자동으로 프로그램과 서비스 요청에 할당할 수 있습니다. Chip 수준 가상화를 사용하면 Intel VT와 AMD Pacifica 하드웨어를 사용한 운영 체제를 hypervisor에서 실행할 수 있습니다.

17.2. 오픈 연결 창

This window appears first and prompts the user to choose a hypervisor session. Non-privileged users can initiate a read-only session. Root users can start a session with full blown read-write status. For normal use, select the **Local Xen host** option. You start the Virtual Machine Manager test mode by selecting the **Other hypervisor** and then type **test:///default** in the URL field beneath. Once in test mode, you can connect to a libvirt dummy hypervisor. Note that although the **Remote Xen host screen** is visible, the functionality to connect to such a host is not implemented into Red Hat Enterprise Linux 5.1.

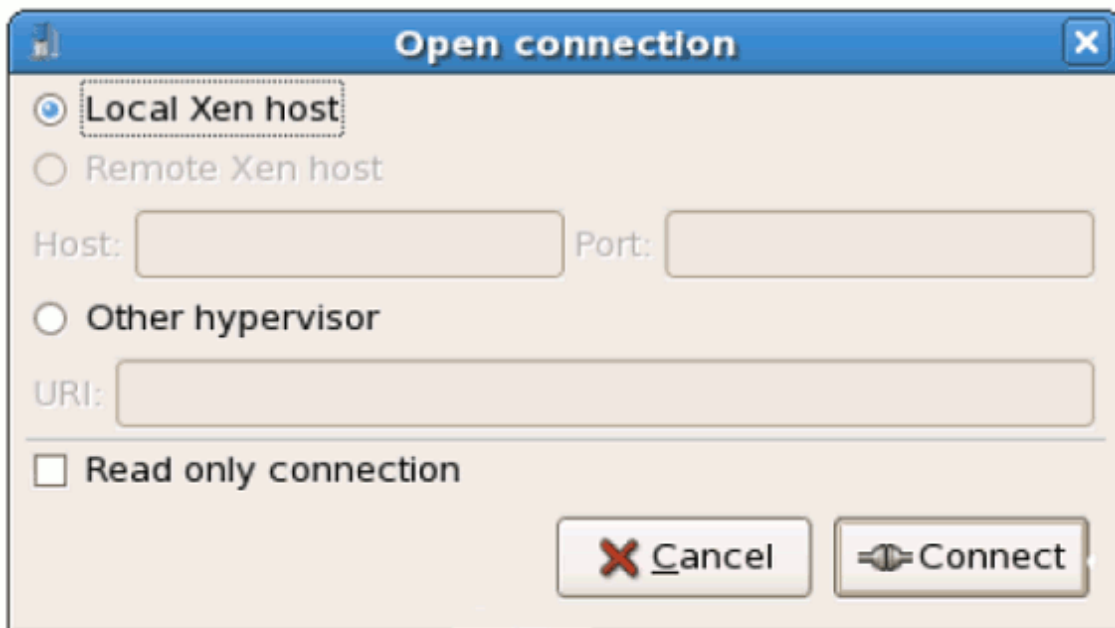


그림 17.1. Virtual Machine Manager Connection window

17.3. 가상 장치 관리자 창

메인 창은 실행 중인 모든 가상 장치와 가상 장치에 현재 할당된 자원을 보여줍니다 (domain0 포함). 사용자는 어느 영역을 표시할지 선택할 수도 있습니다. 원하는 가상 장치에 두 번 클릭하면 특정 장치에 대한 각 콘솔을 보여줍니다. 가상 장치를 선택하고 **상세정보** 버튼을 두 번 클릭하면 장치에 대한 상세정보 창이 표시됩니다. 또한, **파일** 메뉴에서 새로운 가상 장치를 생성할 수도 있습니다.

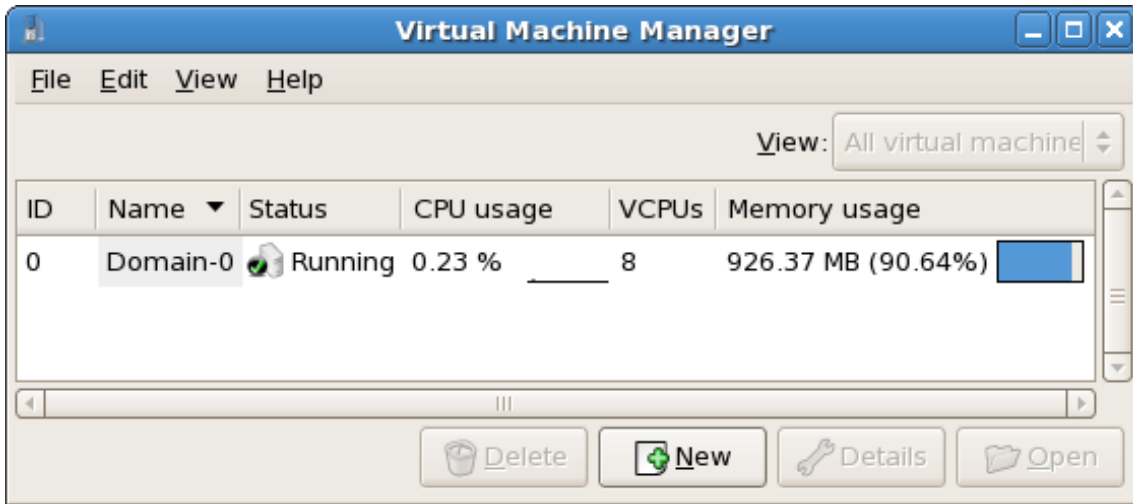


그림 17.2. Virtual Machine Manager main window

17.4. 가상 장치 상세정보 창

This window displays graphs and statistics of a guest's live resource utilization data available from the Red Hat Virtualization Virtual Machine Manager. The UUID field displays the globally unique identifier for the virtual machines(s).

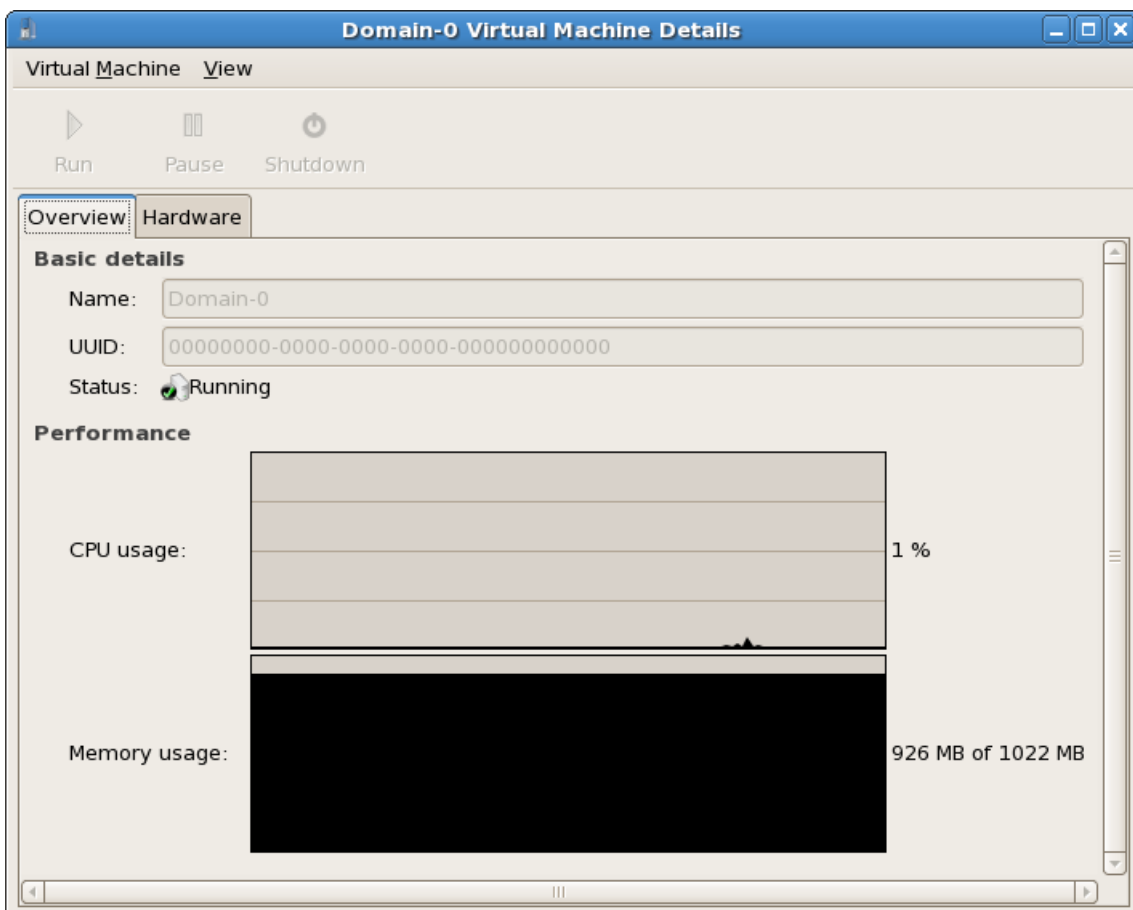


그림 17.3. Virtual Machine Manager Details window

17.5. 가상 장치 그래픽형식 콘솔

This window displays a virtual machine's graphical console. Paravirtual and full virtual machines use different techniques to export their local virtual framebuffers, but both technologies use VNC to make

them available to the Virtual Machine Manager's console window. If your virtual machine is set to require authentication, the Virtual Machine Graphical console prompts you for a password before the display appears.

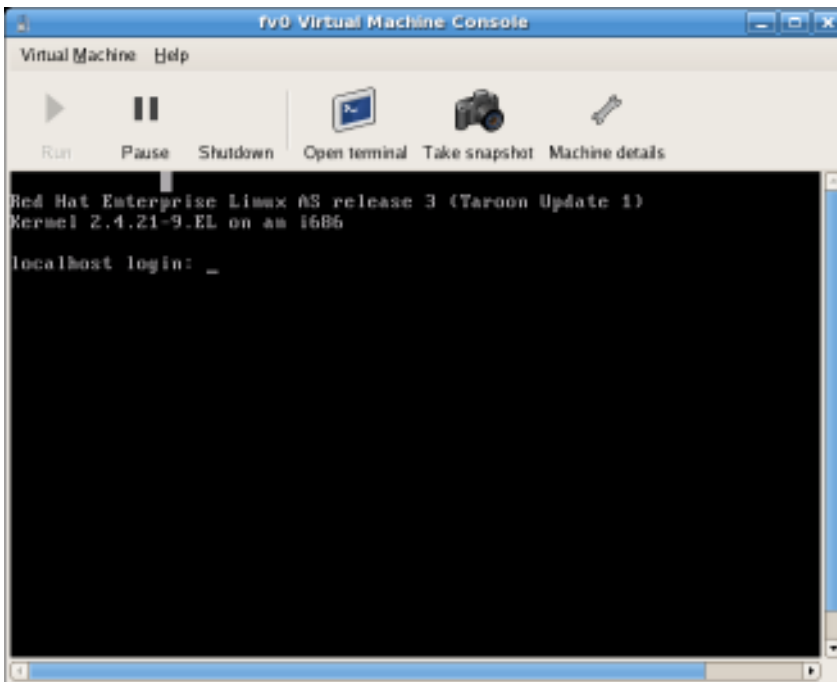


그림 17.4. 그래픽형식 콘솔 창

Your local desktop can intercept key combinations (for example, Ctrl+Alt+F11) to prevent them from being sent to the guest machine. You can use the Virtual Machine Manager's 'sticky key' capability to send these sequences. You must press any modifier key (like Ctrl or Alt) 3 times and the key you specify gets treated as active until the next non-modifier key is pressed. Then you can send Ctrl-Alt-F11 to the guest by entering the key sequence 'Ctrl Ctrl Ctrl Alt+F1'.

17.6. STARTING THE VIRTUAL MACHINE MANAGER

가상 장치 관리자 세션을 시작하려면, 프로그램 메뉴에서 시스템 도구를 클릭하고 **Virtual Machine Manager**를 선택하시기 바랍니다.

가상 장치 관리자 메인 창이 나타납니다.

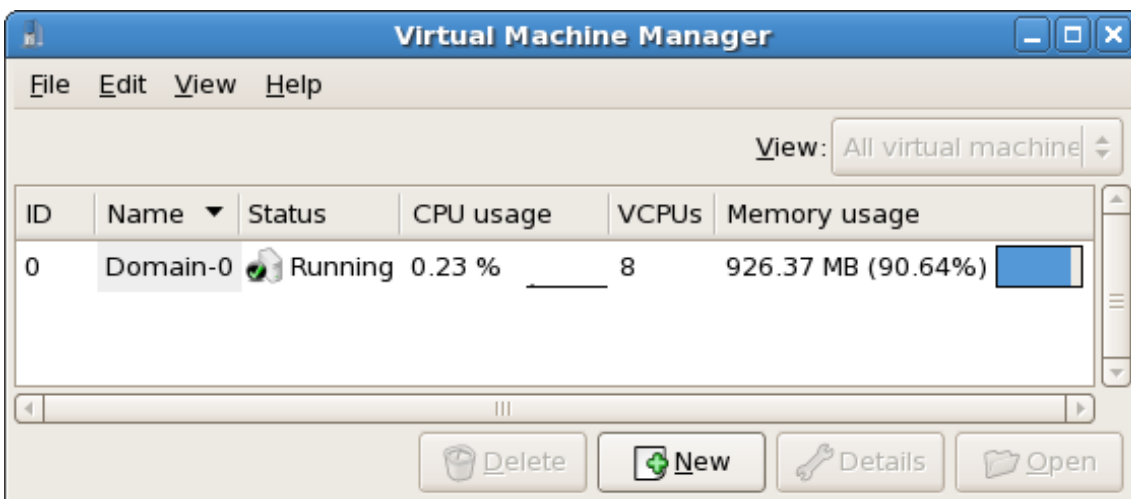


그림 17.5. Starting the Virtual Machine Manager

17.7. 새로운 가상 장치 생성

가상 장치 관리자는 가상 장치를 관리하는 데스크톱 프로그램입니다.

You can use Red Hat's Virtual Machine Manager to:

- 새로운 도메인을 생성합니다.
- Configure or adjust a domain's resource allocation and virtual hardware.
- 실시간 성능과 자원 활용 통계를 이용해 실행 중인 도메인을 요약합니다.
- 매시간 성능과 자원 활용을 나타내는 그래프를 표시합니다.
- 게스트 도메인에 대한 완전 그래픽형식 콘솔을 보여주는 내장된 VNC 클라이언트 보기 프로그램을 사용합니다.



참고

You must install Red Hat Enterprise Linux 5.1, virt-manager, and the kernel packages on all systems that require virtualization. All systems then must be booted and running the Red Hat Virtualization kernel.

다음은 가상 장치 감시를 사용하는 Red Hat Enterprise Linux 5에 게스트 운영 체제를 설치하는 단계를 보여줍니다:

절차 17.1. 게스트 운영 체제 생성

1. 프로그램 메뉴에서 시스템 도구와 **Virtual Machine Manager**를 선택하시기 바랍니다.

가상 장치 관리자 메인 창이 나타납니다.

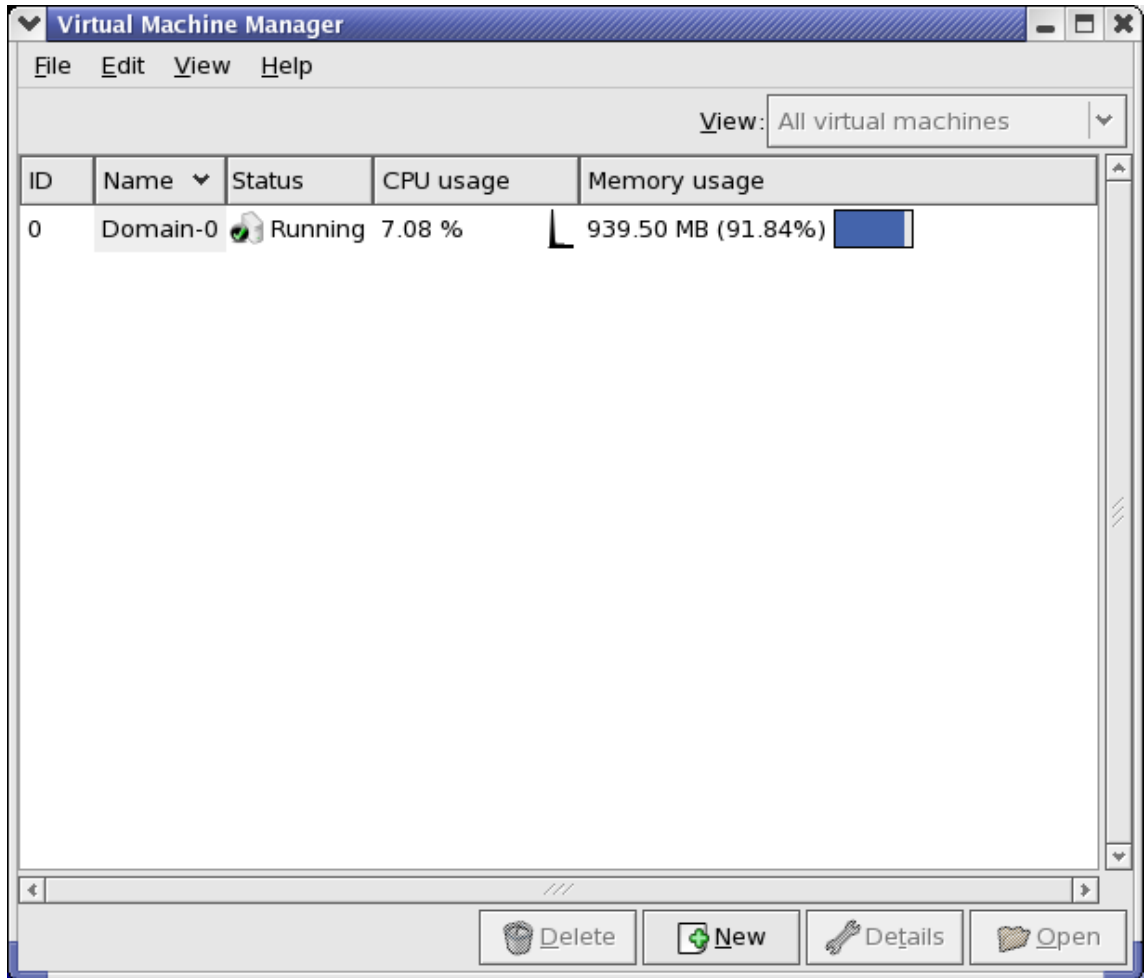


그림 17.6. Virtual Machine Manager window

2. 파일 메뉴에서 새 장치를 선택하시기 바랍니다.

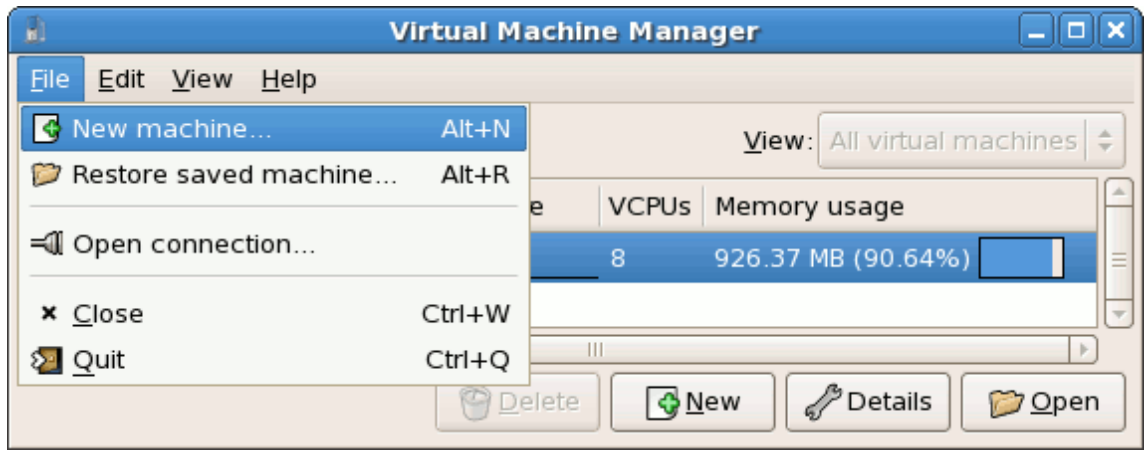


그림 17.7. 새 장치 선택

새 가상 장치 마법사가 나타납니다.

3. Click **Forward**.

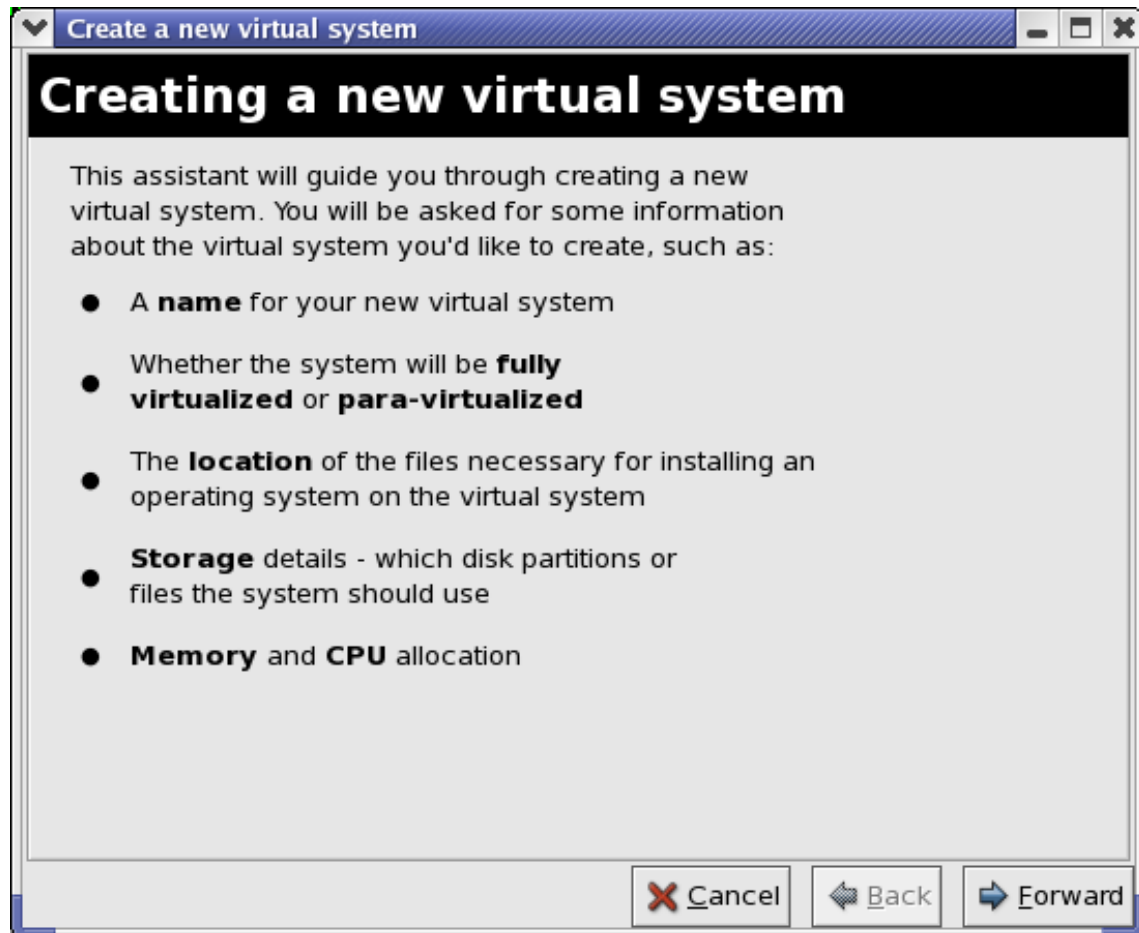


그림 17.8. 새 가상 시스템 마법사 생성

4. Enter the name of the new virtual system and then click **Forward**.

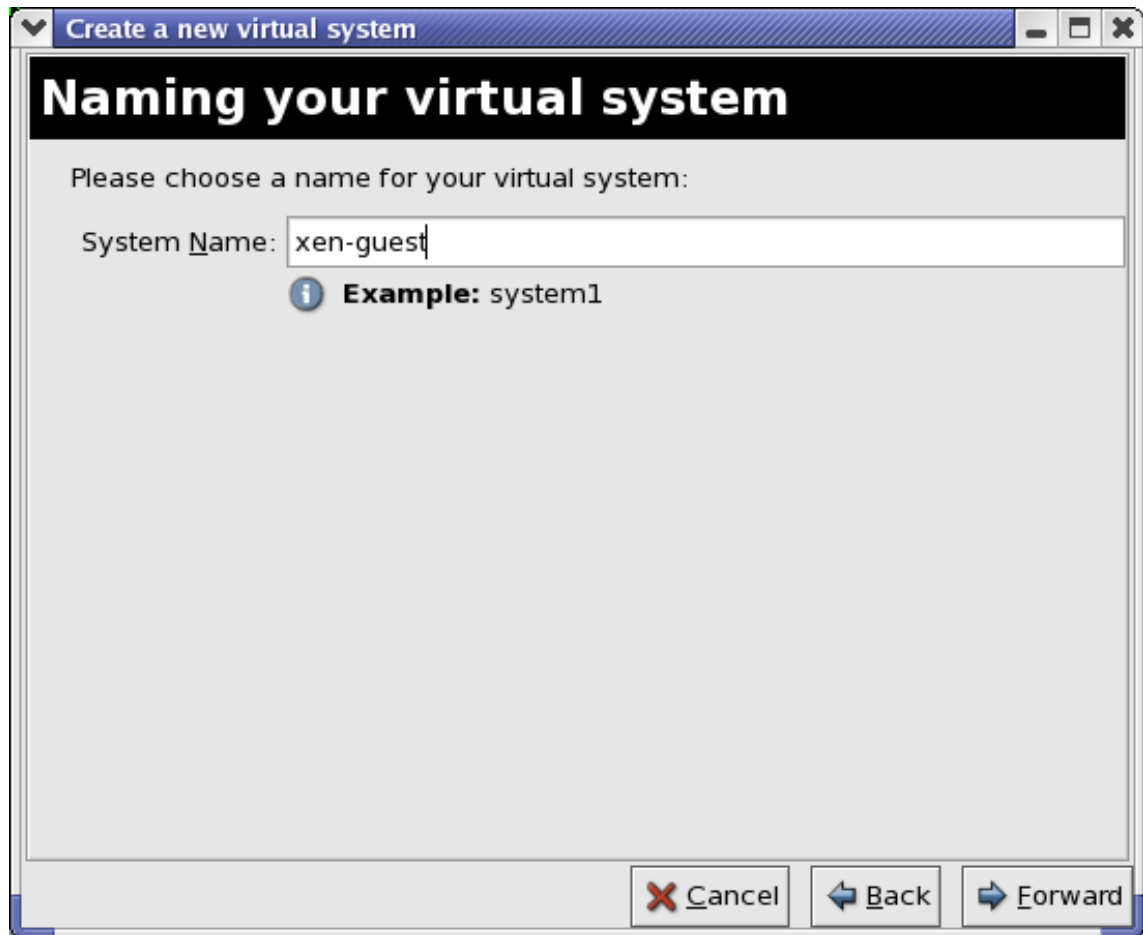


그림 17.9. 가상 시스템 이름 짓기

5. 설치 개체의 위치를 입력하고 **앞으로**를 클릭합니다 (선택적으로 키스타트 파일 위치를 입력할 수 있습니다).

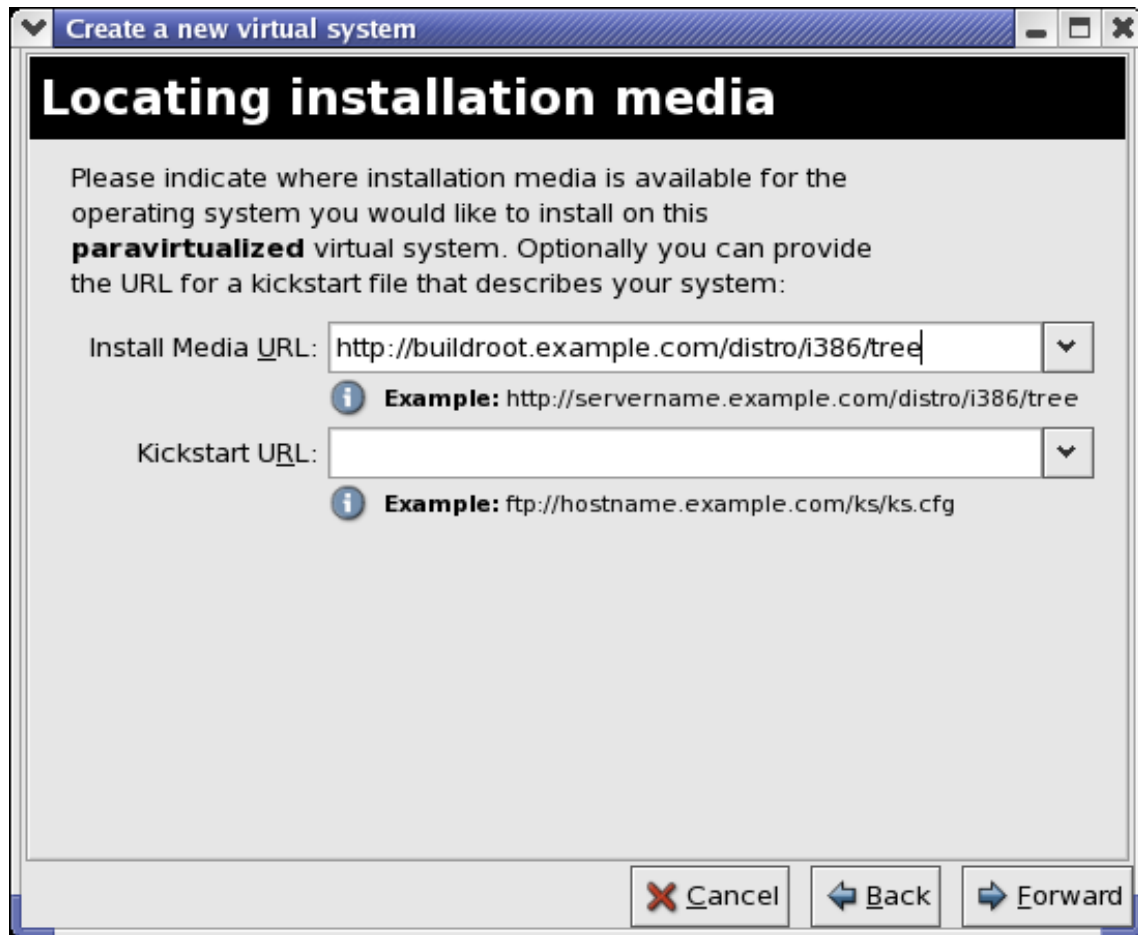


그림 17.10. 설치 매체 위치

6. 물리 디스크 파티션 또는 파일을 사용하여 가상 파일 시스템으로 설치합니다.



참고

이 예제는 파일을 사용하여 가상 시스템을 설치합니다.

SELinux policy only allows xen disk images to reside in `/var/lib/xen/images`.

Open a terminal and create the `/xen` directory and set the SELinux policy with the command `restorecon -v /xen`. Specify your location and the size of the virtual disk, then click **Forward**.



그림 17.11. 저장 공간 할당

7. Select memory to allocate the guest and the number of virtual CPUs then click **Forward**.

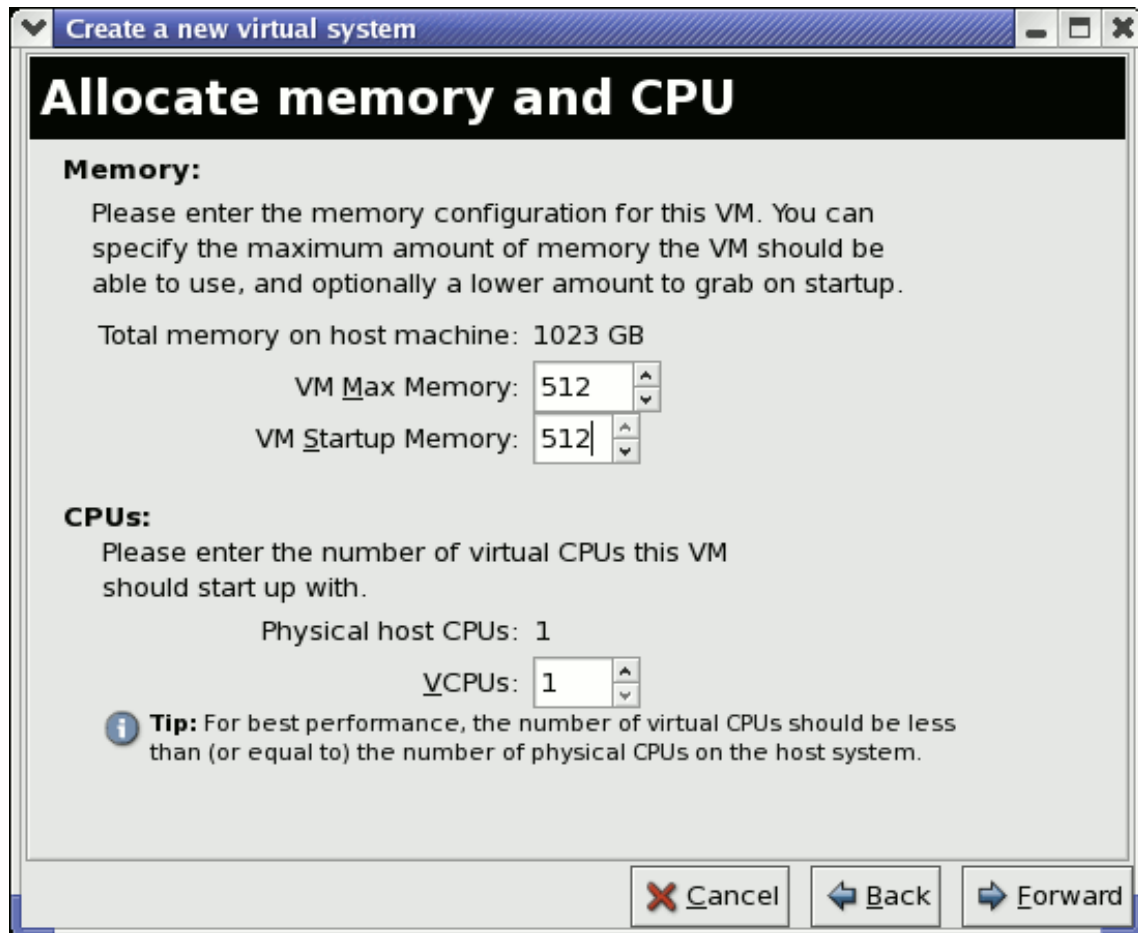


그림 17.12. 메모리 및 CPU 할당

8. Select **Forward** to open a console and the files start to install.

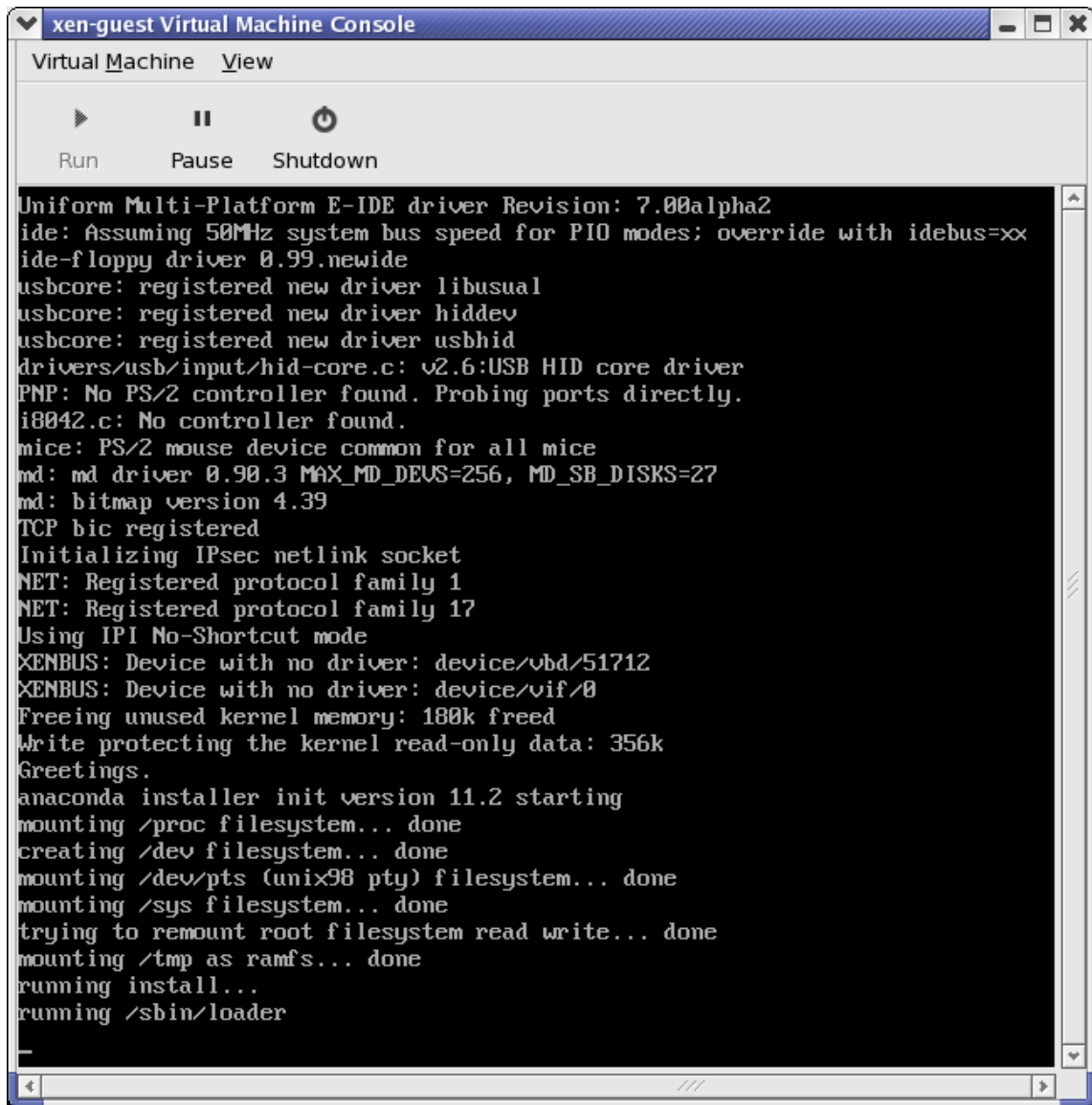


그림 17.13. 메모리 및 CPU 할당

9. 화면에 제공되는 창에서 설치를 완료합니다.

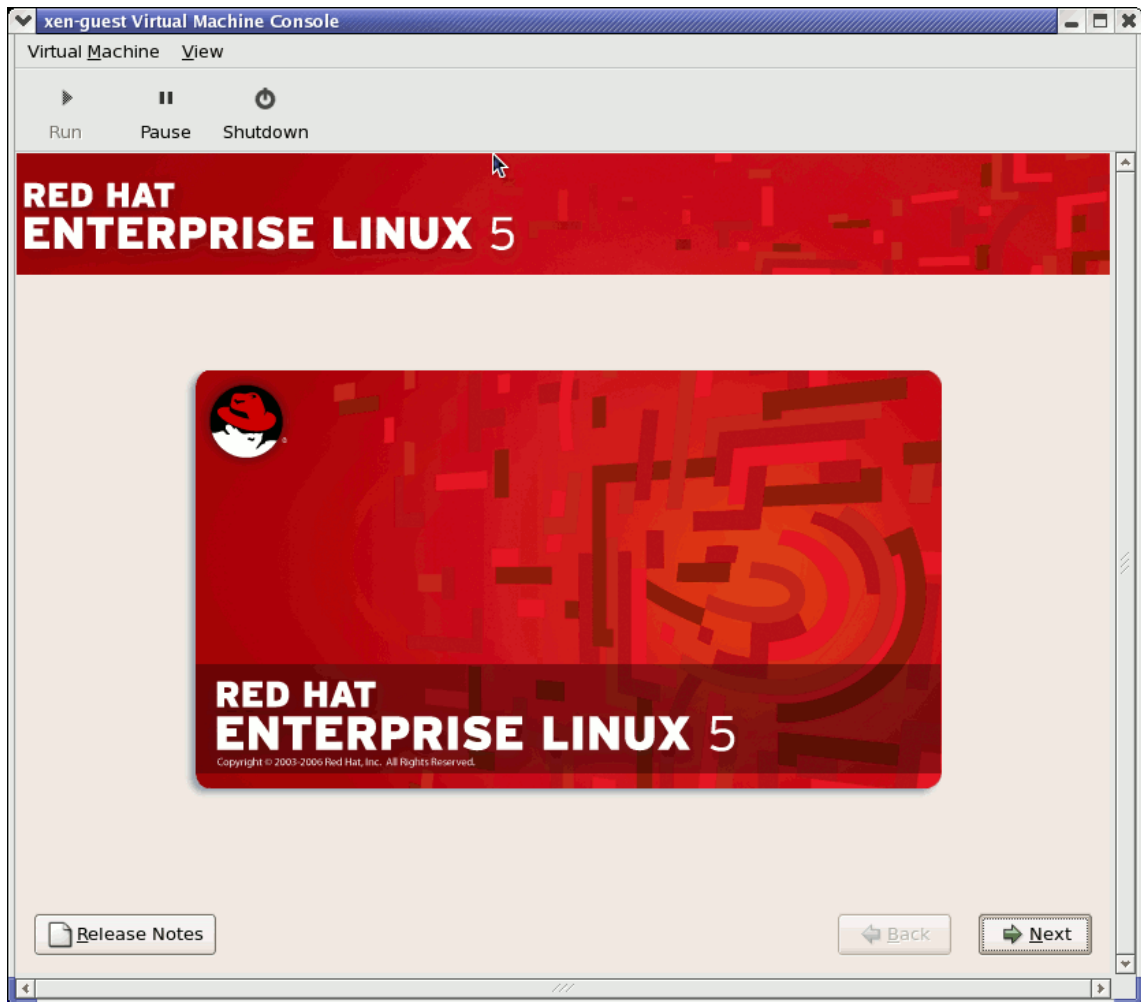


그림 17.14. 설치 시작...



주의

When installing Red Hat Enterprise Linux 5.1 on a fully virtualized guest, do *not* use the **kernel-xen** kernel. Using this kernel on fully virtualized guests can cause your system to hang.

If you are using an *Installation Number* when installing Red Hat Enterprise Linux 5.1 on a fully virtualized guest, be sure to deselect the **Virtualization** package group during the installation. The **Virtualization** package group option installs the **kernel-xen** kernel.

Note that paravirtualized guests are not affected by this issue. Paravirtualized guests always use the **kernel-xen** kernel.

10. Type `xm create -c xen-guest` to start the Red Hat Enterprise Linux 5.1 guest. Right click on the guest in the Virtual Machine Manager and choose **Open** to open a virtual console.

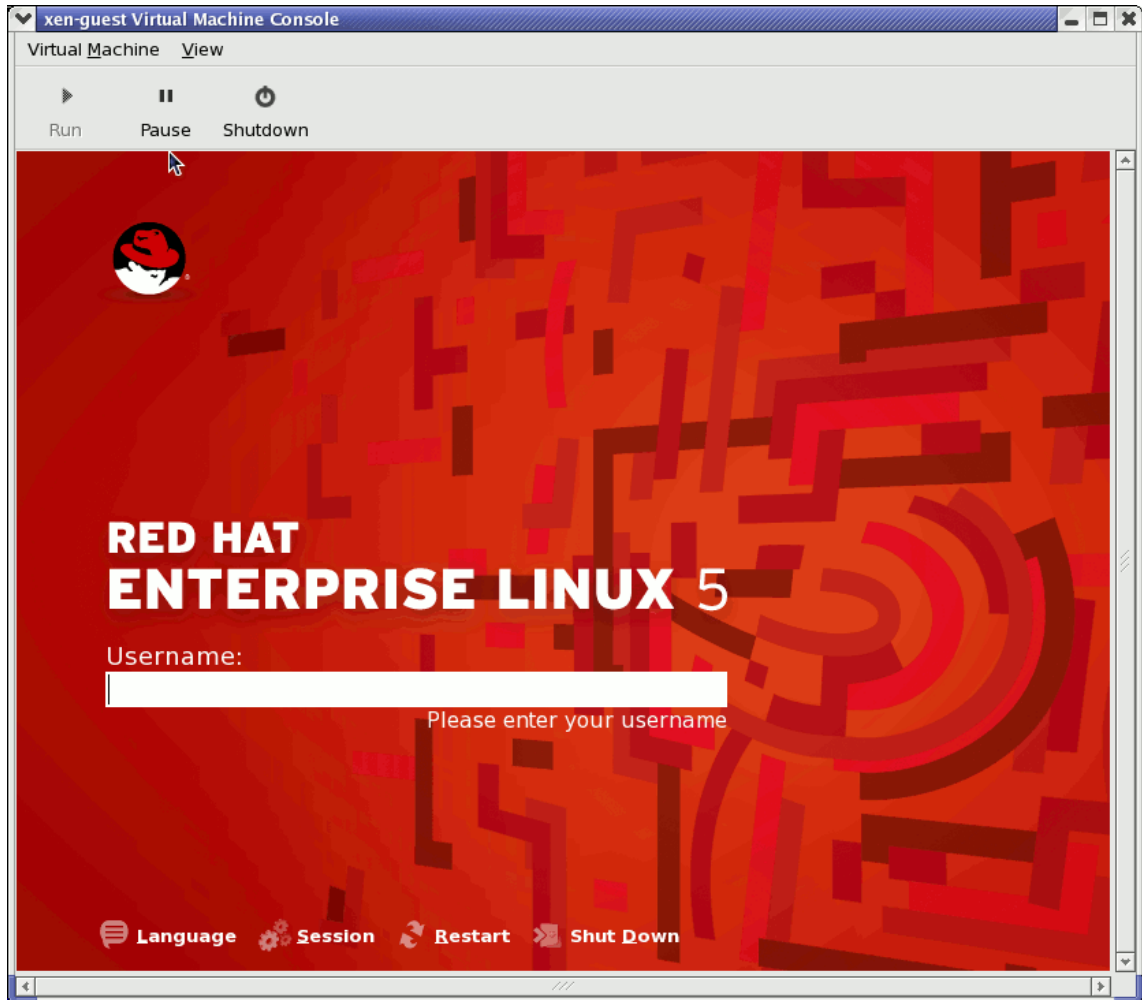


그림 17.15. Red Hat Enterprise Linux 5.1 (guest)

11. 사용자 이름과 패스워드를 넣고 가상 장치 관리자를 계속 사용합니다.

17.8. 저장된 장치 복구

가상 장치 관리자를 시작한 후에, 시스템의 모든 가상 장치는 메인 창에 표시됩니다. `domain0`는 호스트 시스템입니다. 메인 창에 어떤 장치도 나타나지 않으면, 시스템에 현재 실행 중인 장치가 없음을 가리킵니다.

이전에 저장한 세션을 복구하려면, 다음을 수행하시기 바랍니다:

1. 파일 메뉴에서 저장된 장치 복구를 선택합니다.

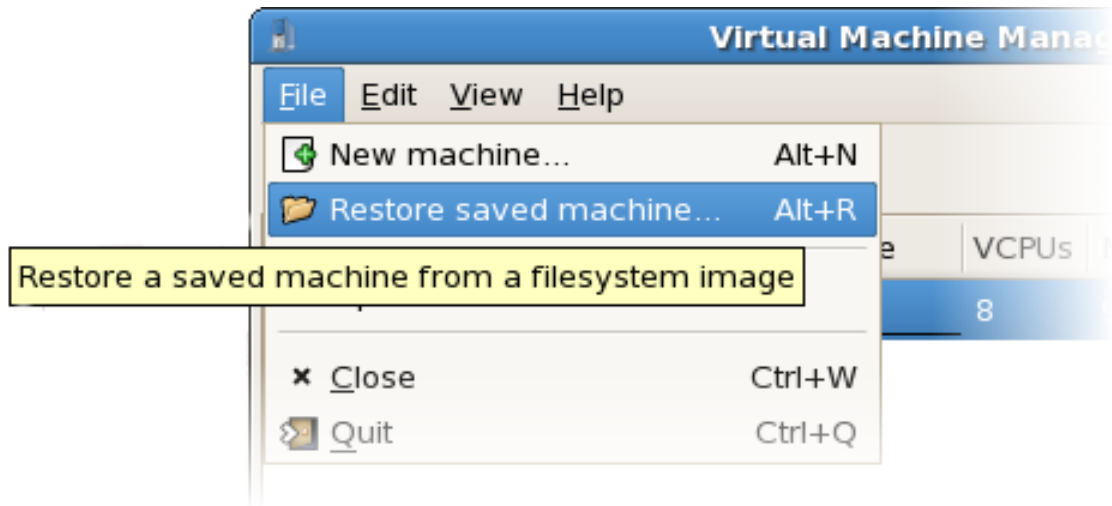


그림 17.16. Restoring a Virtual Machine

2. 가상 장치 복구 메인 창이 나타납니다.

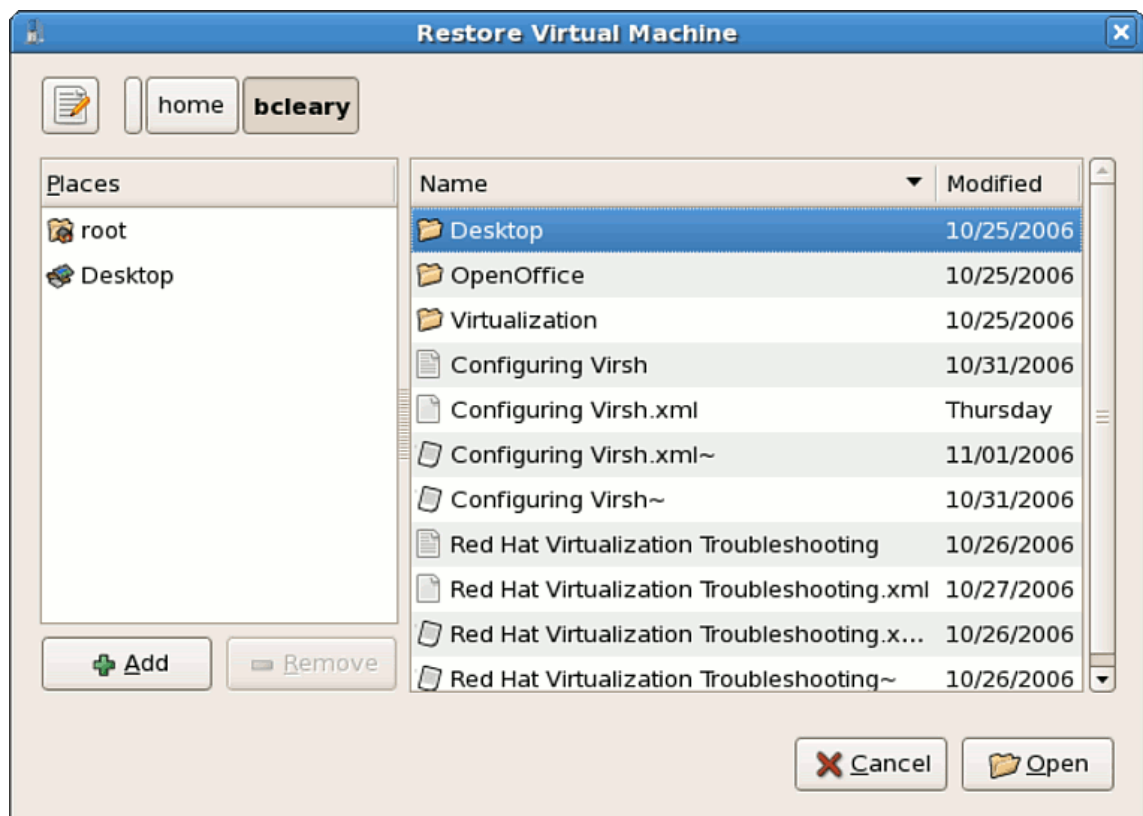


그림 17.17. 저장된 가상 장치 세션 선택

3. 적절한 디렉토리를 탐색하여 저장된 세션 파일을 선택합니다.
4. 열기 클릭합니다.

저장된 가상 시스템은 이제 가상 장치 관리자 메인 창에 나타납니다.

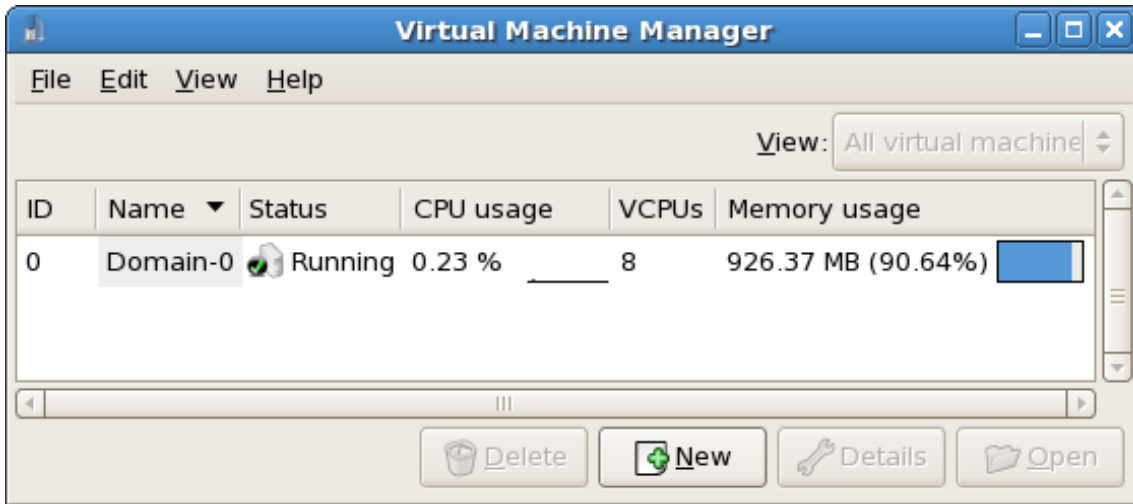


그림 17.18. 복구된 가상 장치 관리자 세션

17.9. DISPLAYING VIRTUAL MACHINE DETAILS

가상 장치 감시 기능을 사용하여 시스템의 모든 가상 장치의 활성 데이터 정보를 볼 수 있습니다.

To view a virtual system's details:

1. 가상 장치 관리자 메인 창에 보기를 원하는 가상 장치를 하이라이트하시기 바랍니다.

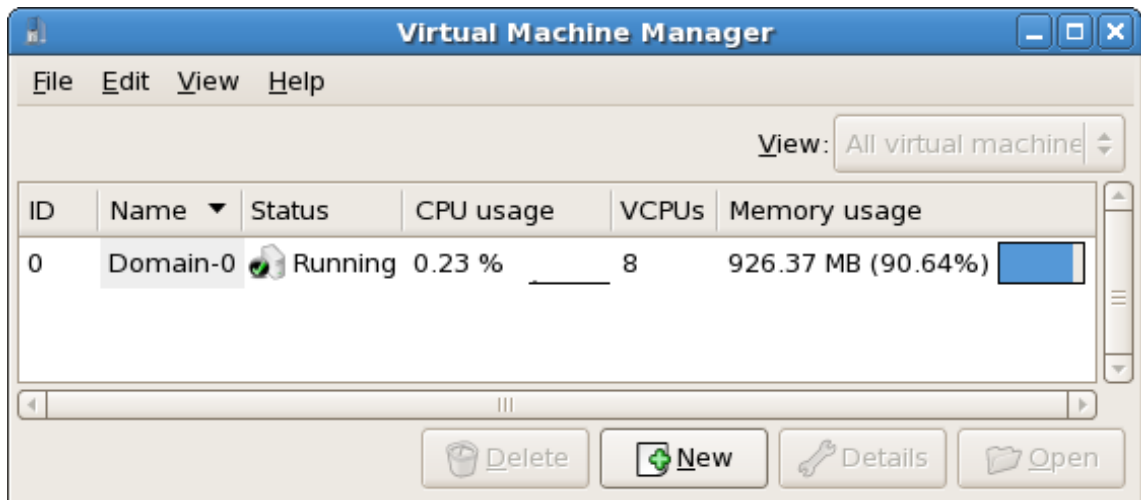


그림 17.19. 표시하려는 가상 장치 선택

2. 가상 장치 관리자의 편집 메뉴에서, 장치 정보를 선택합니다 (또는 가상 장치 관리자의 메인 창 아래에서 상세정보 버튼을 클릭합니다).

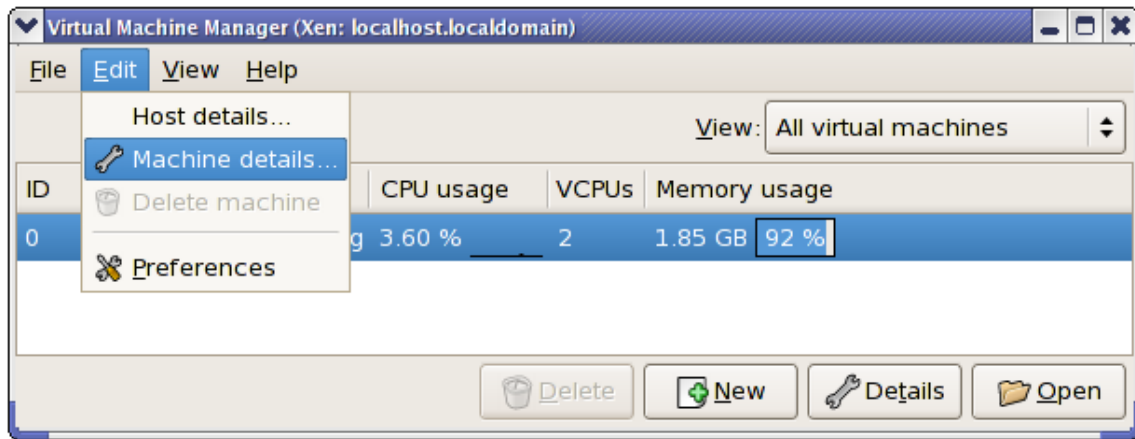


그림 17.20. 가상 장치 상세정보 메뉴 표시

가상 장치 상세정보 개요 창이 나타납니다. 이 창은 지정한 도메인에 사용되는 CPU와 메모리 사용량에 대한 정보를 요약하여 표시합니다.

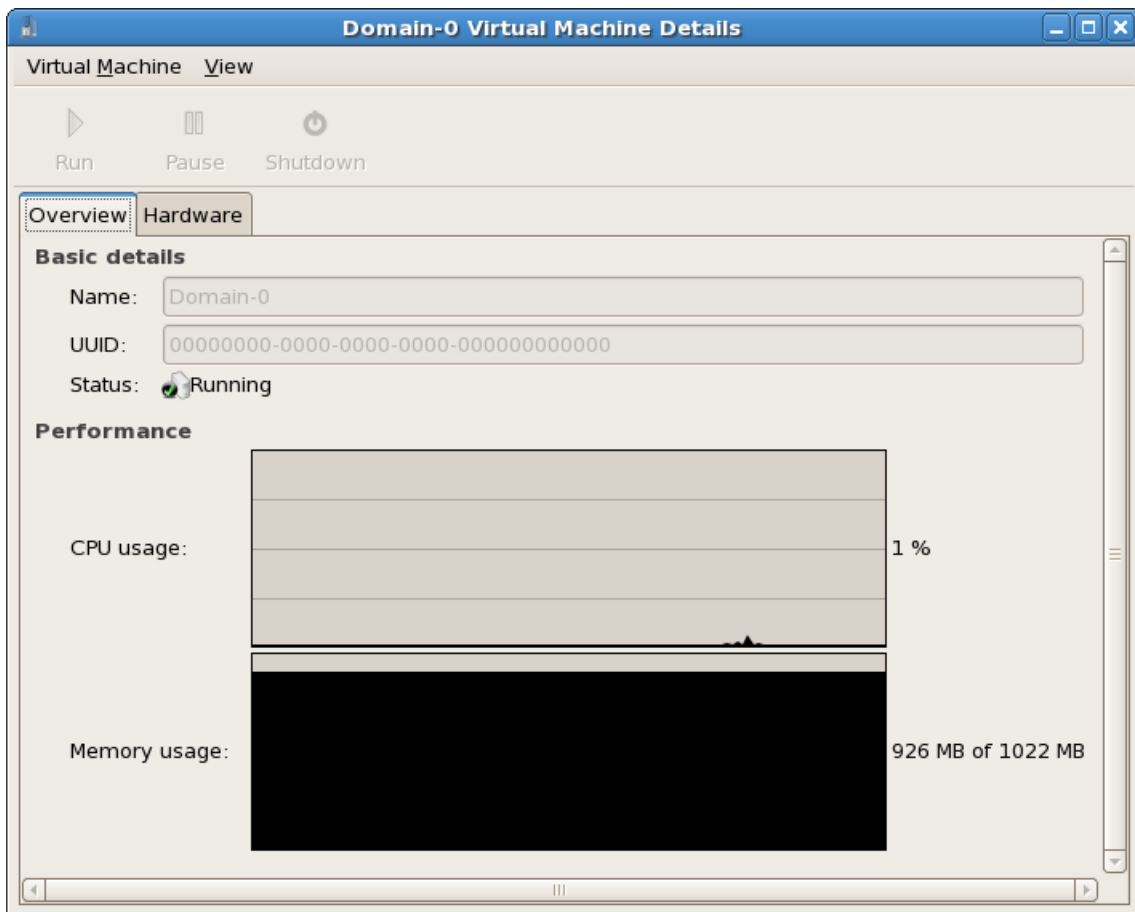


그림 17.21. 가상 장치 상세정보 개요 표시

- 가상 장치 상세정보 창에서 하드웨어 탭을 클릭해 보시기 바랍니다.

가상 장치 상세정보 하드웨어 창이 나타납니다.

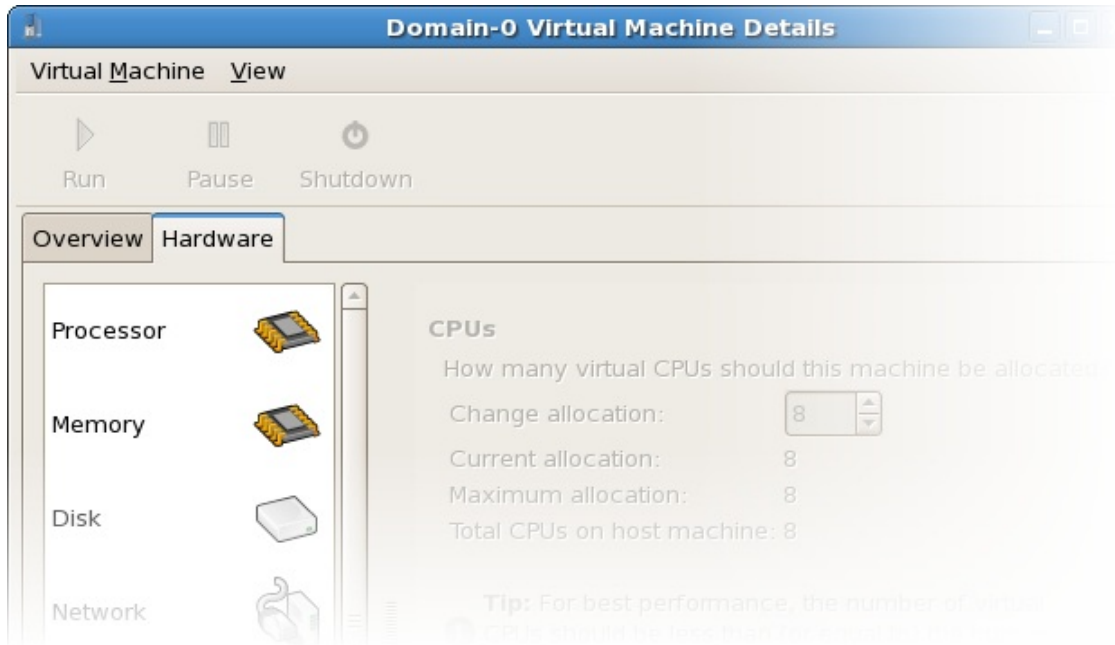


그림 17.22. 가상 장치 상세정보 하드웨어 표시

4. 하드웨어 탭에서 **Processor**를 클릭하면 현재 프로세서 메모리 할당량을 보고 수정할 수 있습니다.



그림 17.23. 프로세서 할당량 표시

5. 하드웨어 탭에서 **Memory**를 클릭하면 현재 RAM 메모리 할당량을 보고 수정할 수 있습니다.

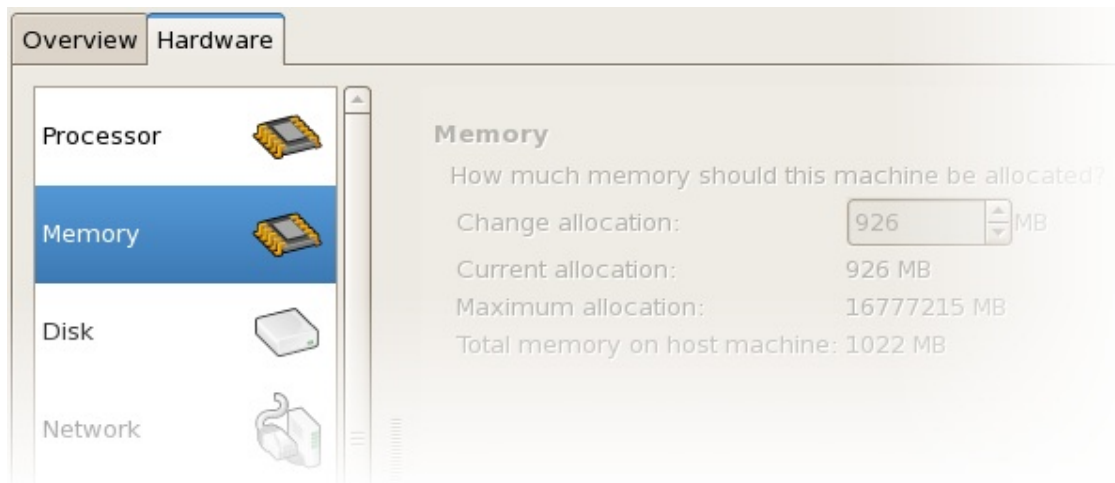


그림 17.24. 메모리 할당량 표시

6. 하드웨어 탭에서 **Disk**를 클릭하면 현재 하드 디스크 구성을 보고 수정할 수 있습니다.

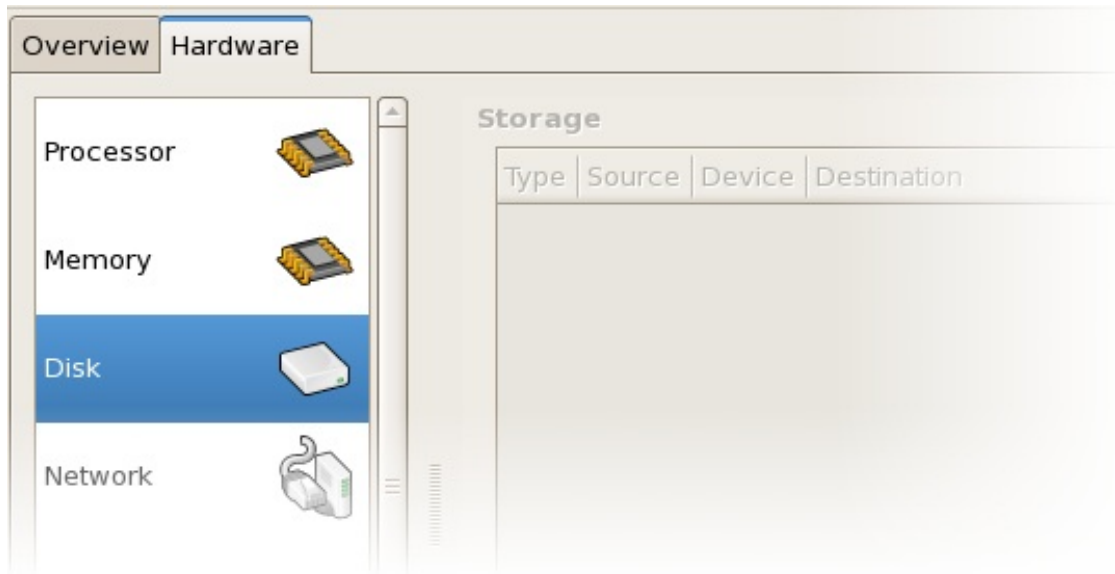


그림 17.25. 디스크 구성 표시

7. 하드웨어 탭에서 **Network**를 클릭하면 현재 네트워크 구성을 보고 수정할 수 있습니다.

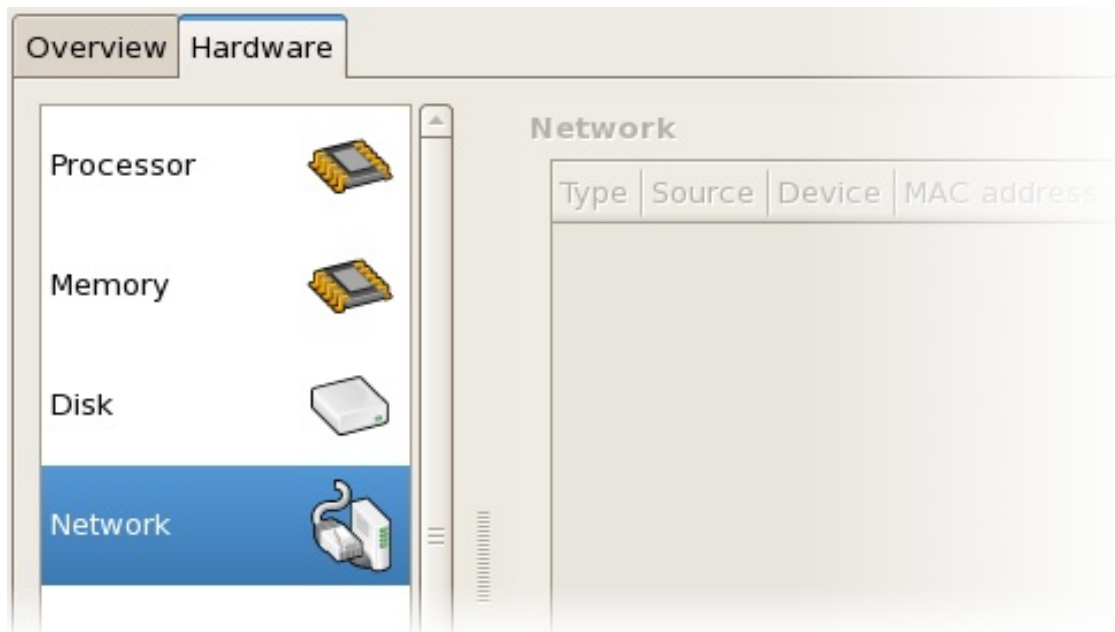


그림 17.26. 네트워크 구성 표시

17.10. CONFIGURING STATUS MONITORING

가상 장치 관리자를 사용하여 가상 시스템의 상태 감시 기능을 수정할 수 있습니다.

상태 감시 기능을 구성하려면, 콘솔을 활성화해야 합니다:

1. 편집 메뉴에서 기본 설정을 선택하시기 바랍니다.

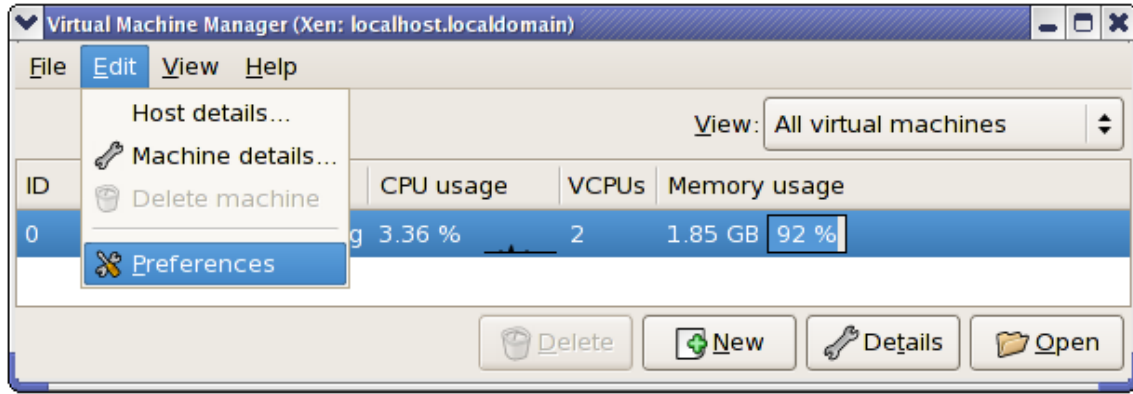


그림 17.27. 가상 장치 등록정보 수정

가상 장치 관리자 등록정보 창이 나타납니다.

2. 상태 감시란의 선택 상자에서, 시스템을 업데이트할 시간(초 단위)을 지정합니다.

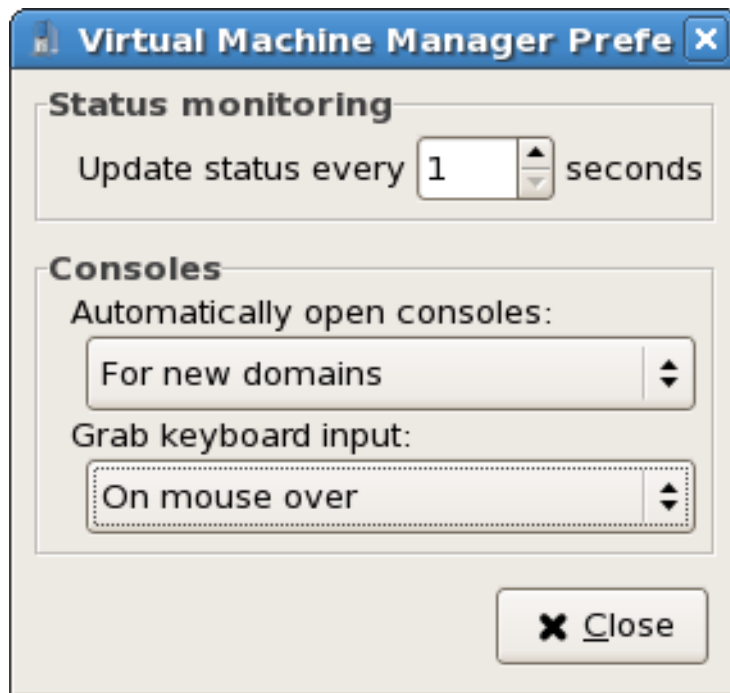


그림 17.28. Configuring Status Monitoring

3. 콘솔란에서, 어떻게 콘솔을 여는지 결정하고 입력 장치를 지정합니다.

17.11. 도메인 ID 표시

시스템의 모든 가상 장치에 대한 도메인 ID를 볼 수 있습니다:

1. 보기 메뉴에서, 도메인 ID 확인 상자를 선택하시기 바랍니다.

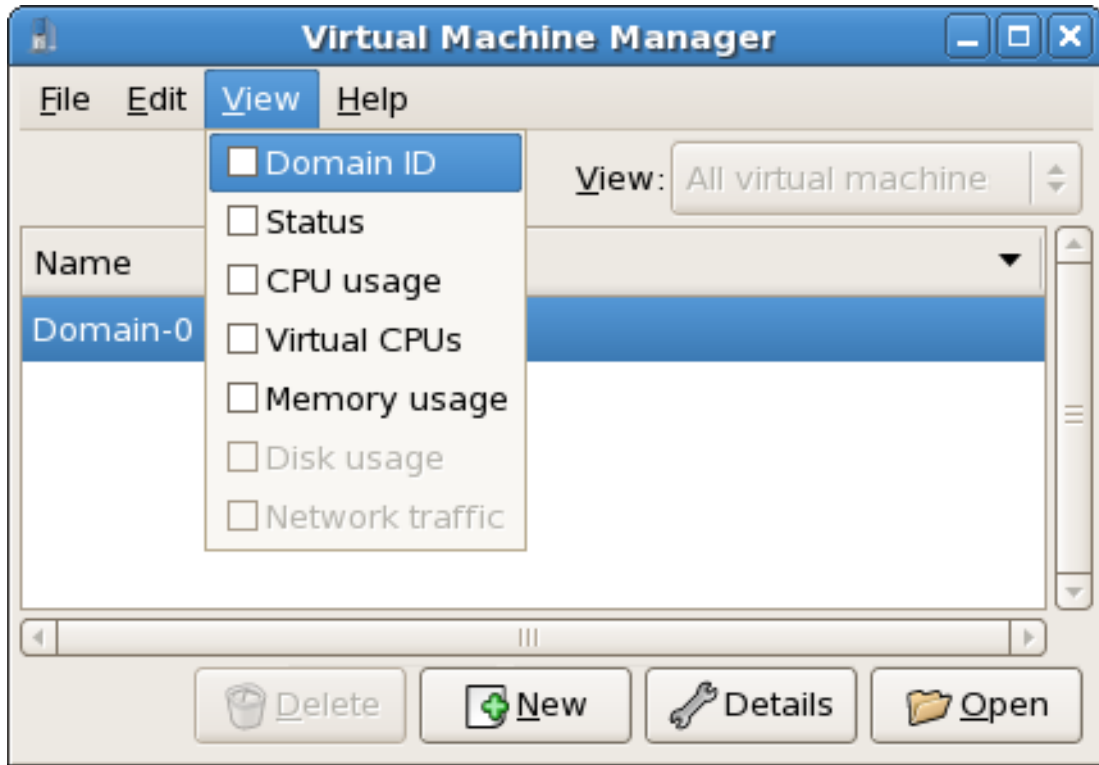


그림 17.29. 도메인 ID 표시

2. The Virtual Machine Manager lists the Domain ID's for all domains on your system.

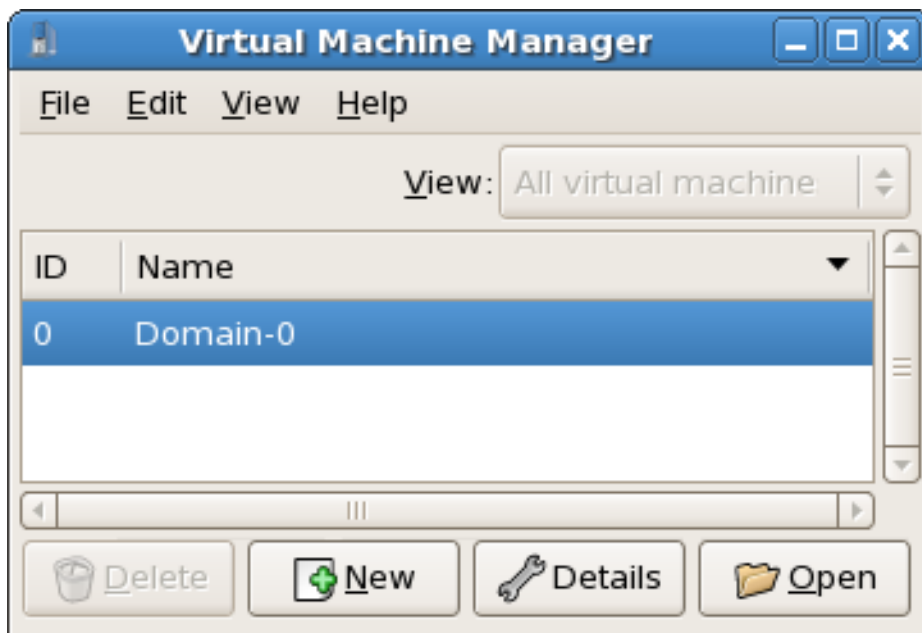


그림 17.30. 도메인 ID 표시

17.12. 가상 장치 상태 표시

시스템의 모든 가상 장치에 대한 상태 정보를 보려면 다음을 수행하시기 바랍니다:

1. 보기 메뉴에서, 상태 확인 상자를 선택합니다.

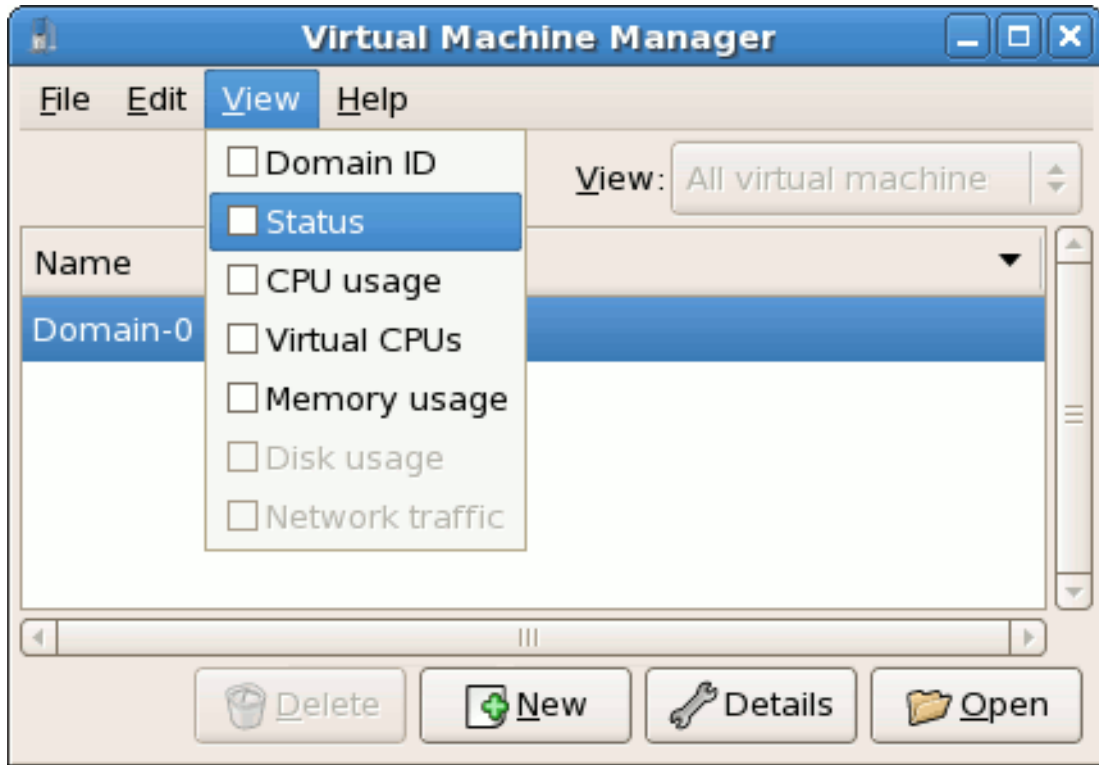


그림 17.31. 가상 장치 상태 표시

- 가상 장치 관리자는 시스템의 모든 가상 장치에 대한 상태 정보를 표시합니다.

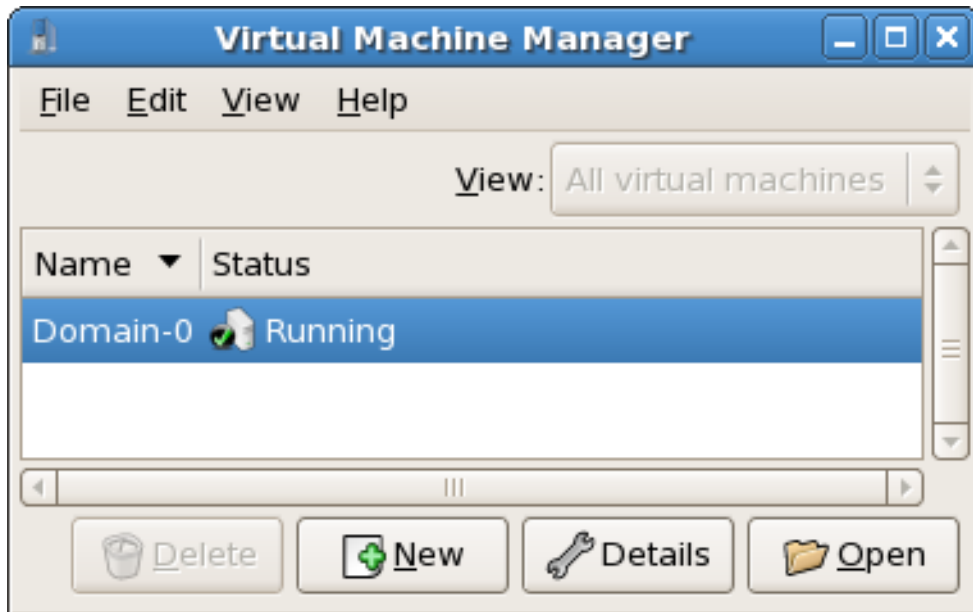


그림 17.32. 가상 장치 상태 표시

17.13. 가상 CPU 표시

시스템의 모든 가상 장치에 대한 가상 CPU 개수를 보려면, 다음을 수행하셔야 합니다:

- 보기 메뉴에서 가상 CPU 확인 상자를 선택하시기 바랍니다.

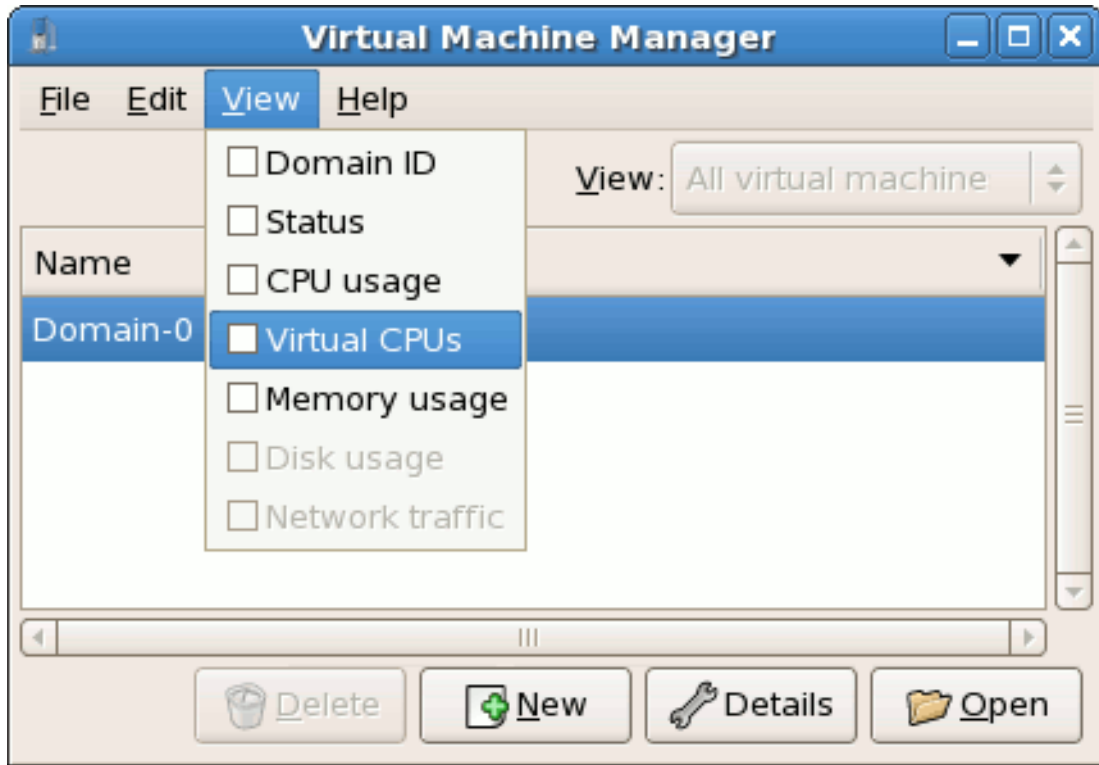


그림 17.33. 가상 CPU 표시

- 가상 장치 관리자는 시스템의 모든 가상 장치에 대한 가상 CPU를 표시합니다.

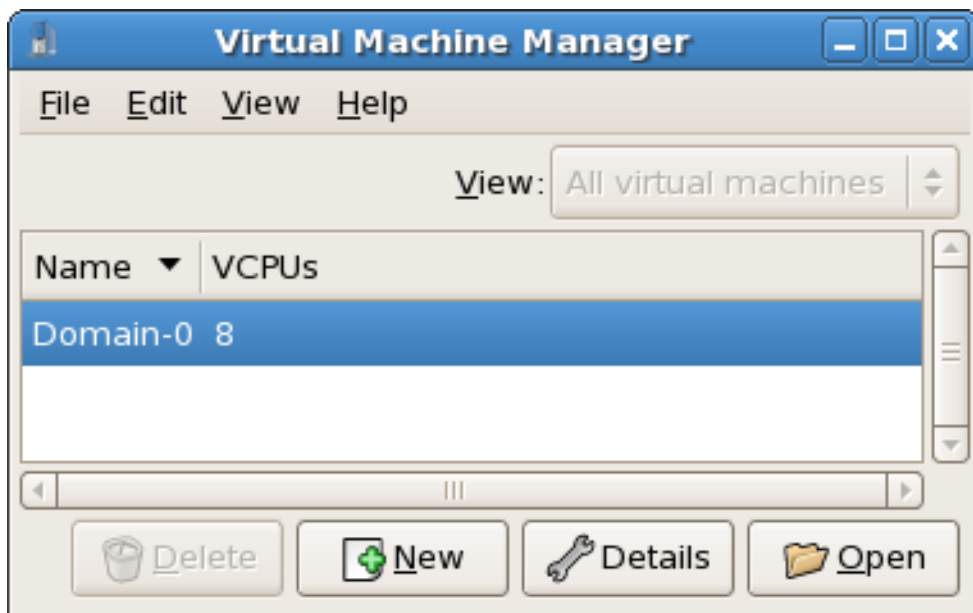


그림 17.34. 가상 CPU 표시

17.14. CPU 사용량 표시

시스템의 모든 가상 장치에 사용되는 CPU 사용량을 볼 수 있습니다:

- 보기 메뉴에서, **CPU 사용량** 확인 상자를 선택하시기 바랍니다.

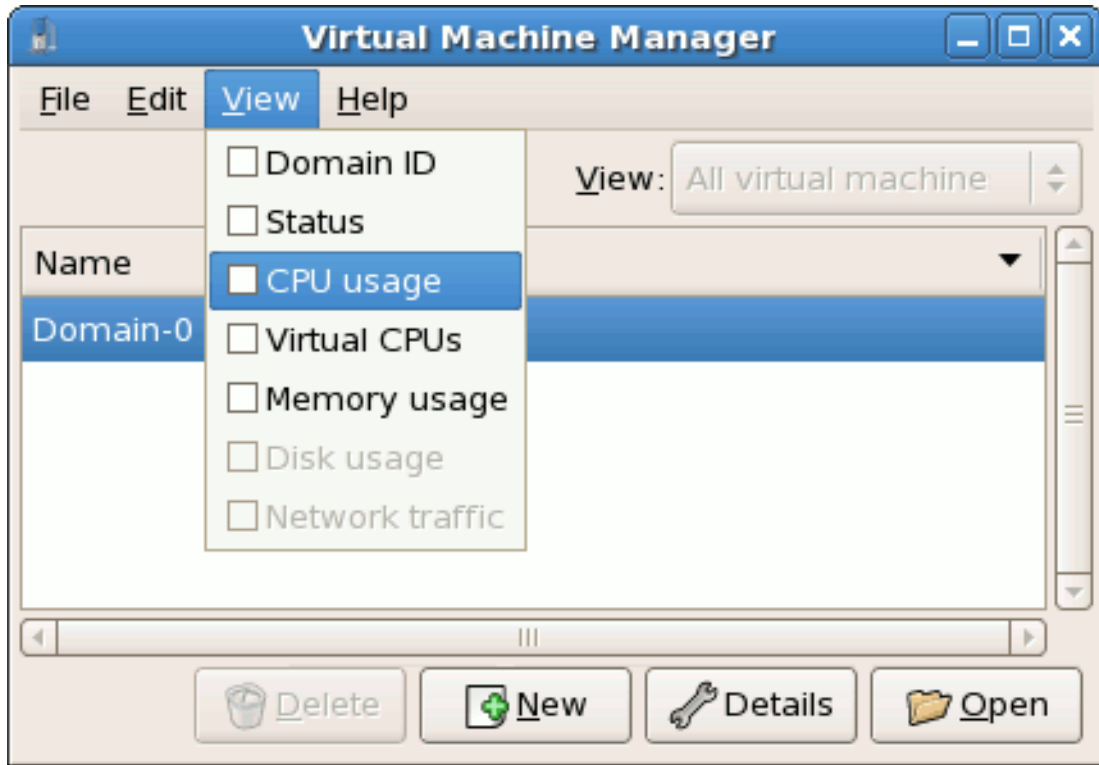


그림 17.35. CPU 사용량 표시

2. 가상 장치 관리자는 시스템의 모든 가상 장치에 사용되는 CPU 사용량을 표시합니다.

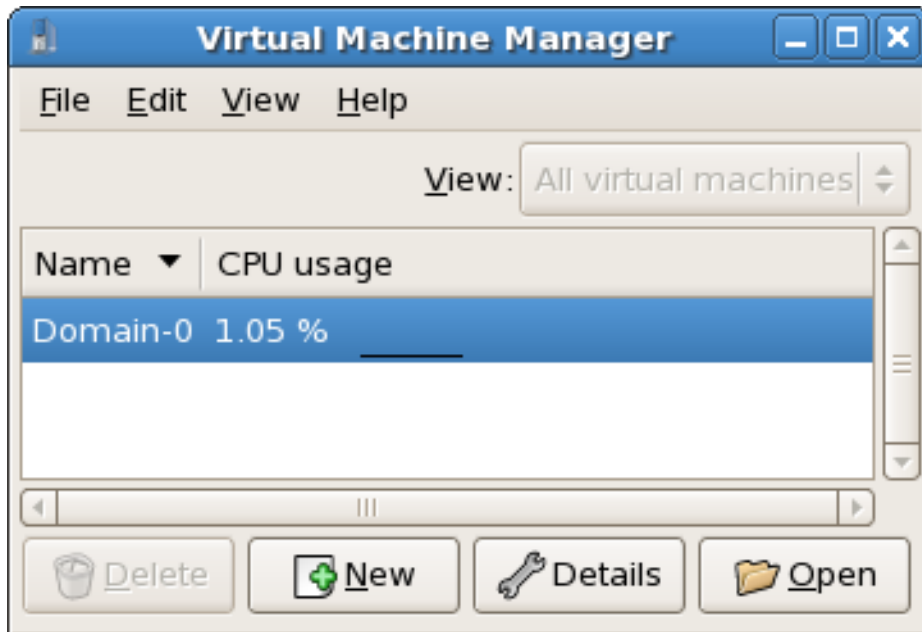


그림 17.36. CPU 사용량 표시

17.15. 메모리 사용량 표시

시스템의 모든 가상 장치에 사용되는 메모리 사용량을 볼 수 있습니다:

1. 보기 메뉴에서, 메모리 사용량 확인 상자를 선택하시기 바랍니다.

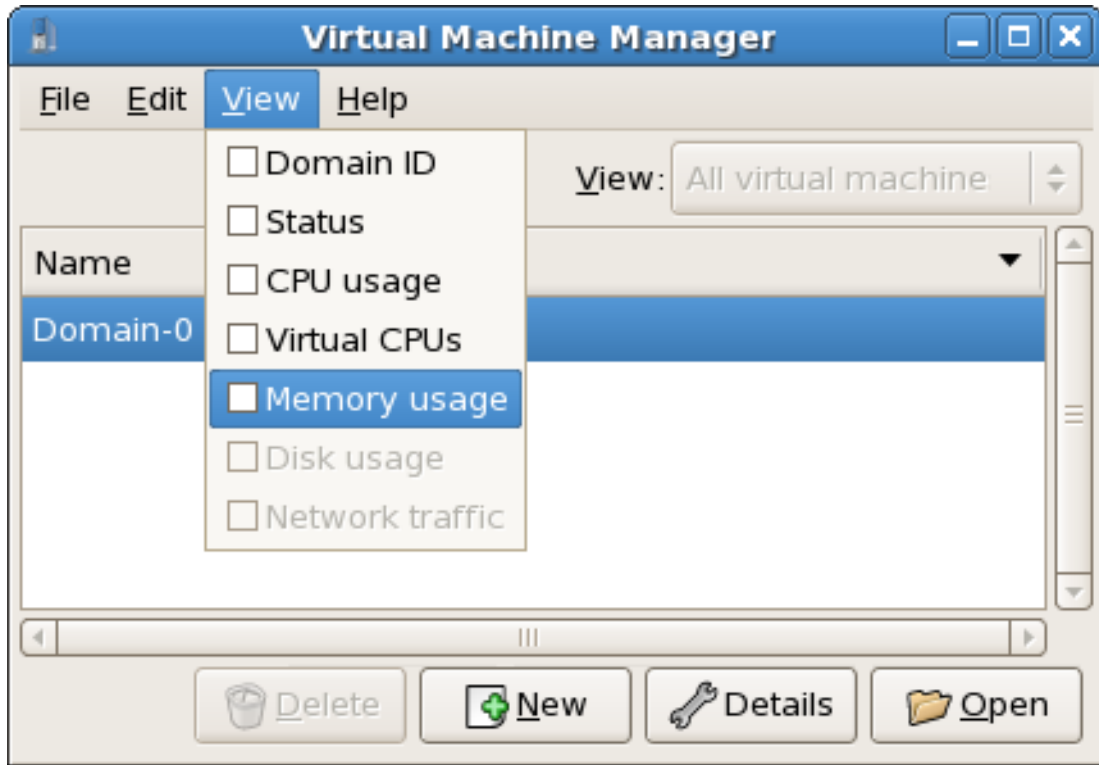


그림 17.37. 메모리 사용량 표시

2. 는 데 장치 관리자는 시스템의 모든 가상 장치에 사용되는 메모리 사용량을 (메가바이트 단위로) 표시합니다.

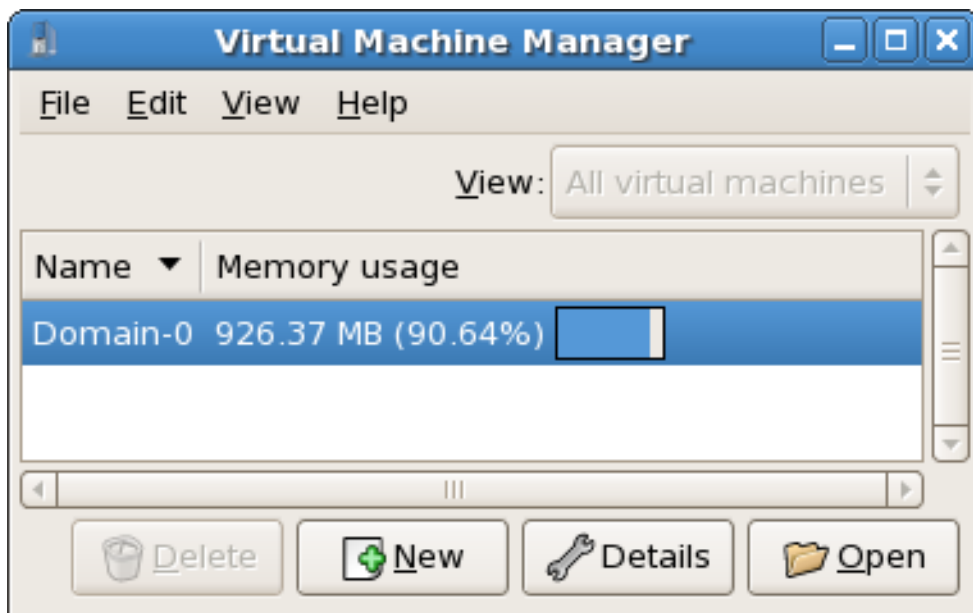


그림 17.38. 메모리 사용량 표시

17.16. MANAGING A VIRTUAL NETWORK

To configure a virtual network on your system:

1. From the **Edit** menu, select **Host Details**.

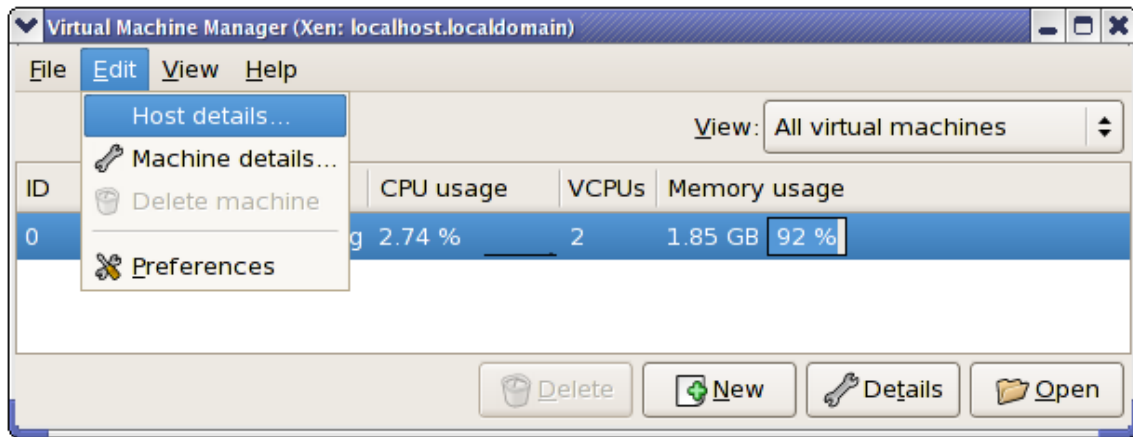


그림 17.39. Selecting Host Details

2. This will open the **Host Details** menu. Click the **Virtual Networks** tab.

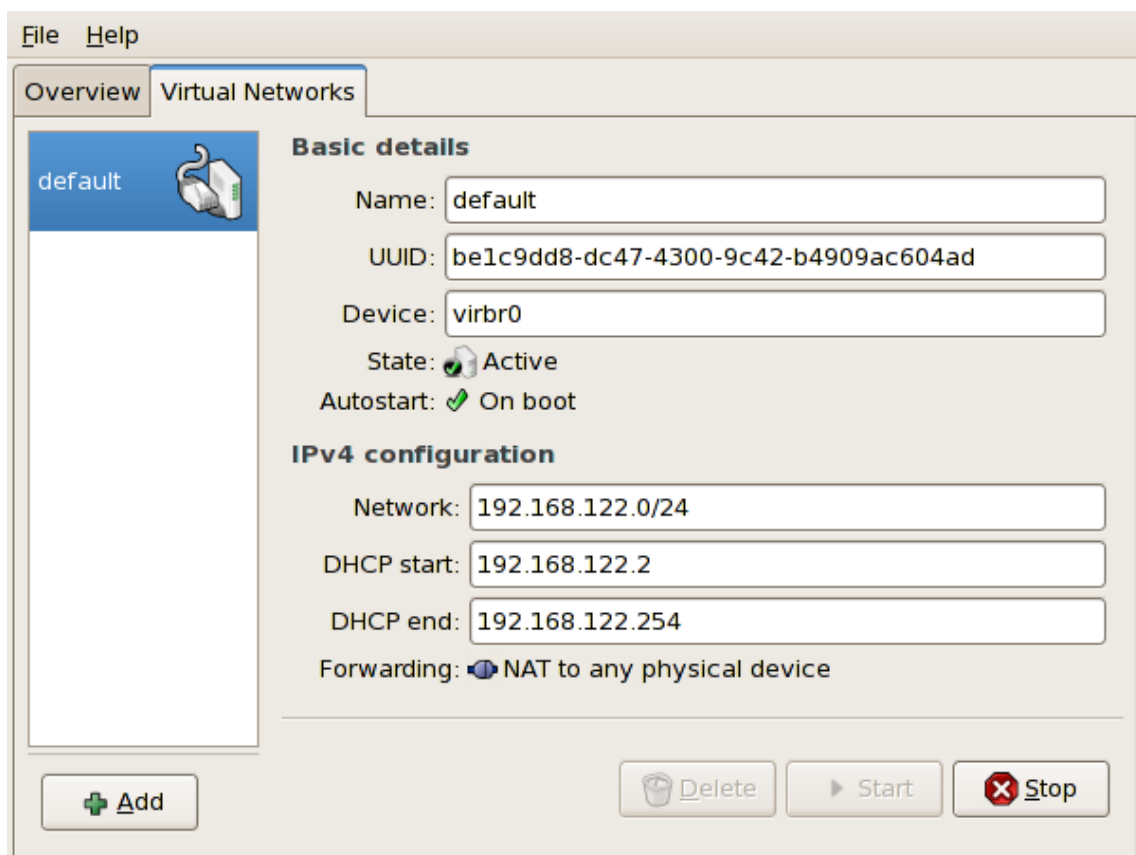


그림 17.40. Virtual Network Configuration

3. All available virtual networks are listed on the left-hand box of the menu. You can edit the configuration of a virtual network by selecting it from this box and editing as you see fit.

17.17. CREATING A VIRTUAL NETWORK

To create a virtual network on your system:

1. Open the **Host Details** menu (refer to 17.16절. "Managing a Virtual Network") and click the **Add** button.

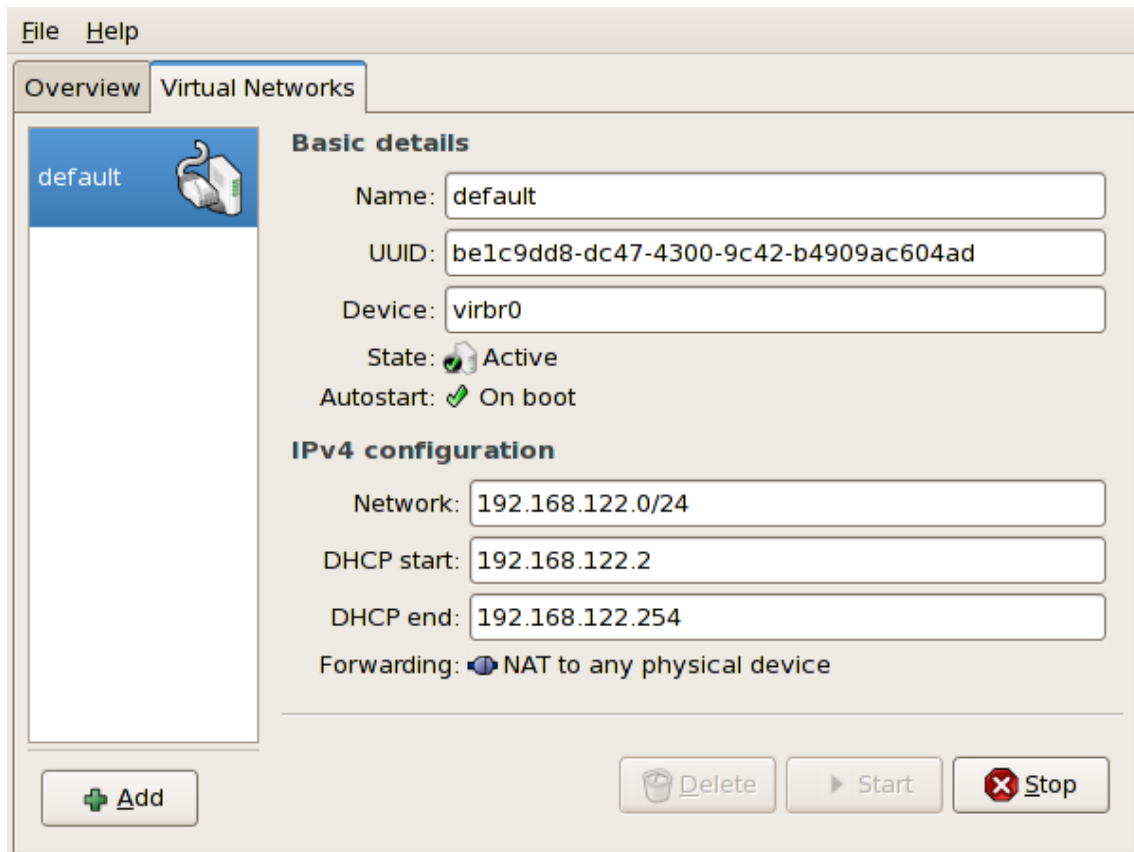


그림 17.41. Virtual Network Configuration

This will open the **Create a new virtual network** menu. Click **Forward** to continue.

Creating a new virtual network

This assistant will guide you through creating a new virtual network. You will be asked for some information about the virtual network you'd like to create, such as:

- A **name** for your new virtual network
- The IPv4 **address** and **netmask** to assign
- The **address range** from which the **DHCP** server will allocate addresses for virtual machines
- Whether to **forward** traffic to the physical network



그림 17.42. Creating a new virtual network

2. Enter an appropriate name for your virtual network and click **Forward**.

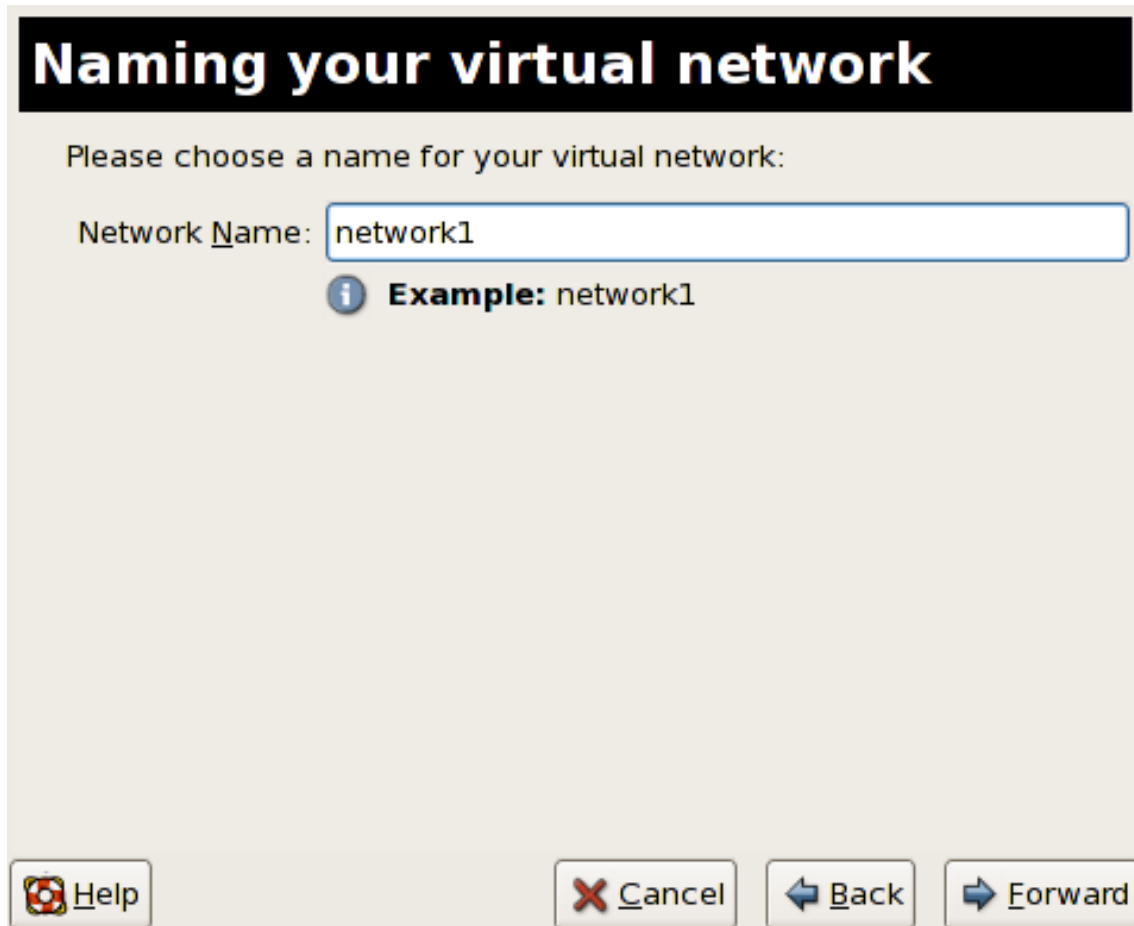



그림 17.43. Naming your virtual network

3. Enter an IPv4 address space for your virtual network and click **Forward**.

Choosing an IPv4 address space

You will need to choose an IPv4 address space for the virtual network:

Network:

 **Hint:** The network should be chosen from one of the IPv4 private address ranges. eg 10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16

Netmask: 255.255.255.0

Broadcast: 192.168.100.255

Gateway: 192.168.100.1

Size: 256 addresses

Type: Private

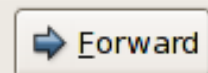
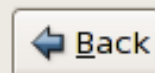
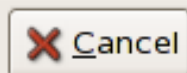
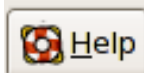


그림 17.44. Choosing an IPv4 address space


4. Define the DHCP range for your virtual network by specifying a **Start** and **End** range of IP addresses. Click **Forward** to continue.

Selecting the DHCP range

Please choose the range of addresses the DHCP server can use to allocate to guests attached to the virtual network

Start:

End:

 **Tip:** Unless you wish to reserve some addresses to allow static network configuration in virtual machines, these parameters can be left with their default values.





 Help  Cancel  Back  Forward

그림 17.45. Selecting the DHCP range

5. Select how the virtual network should connect to the physical network.

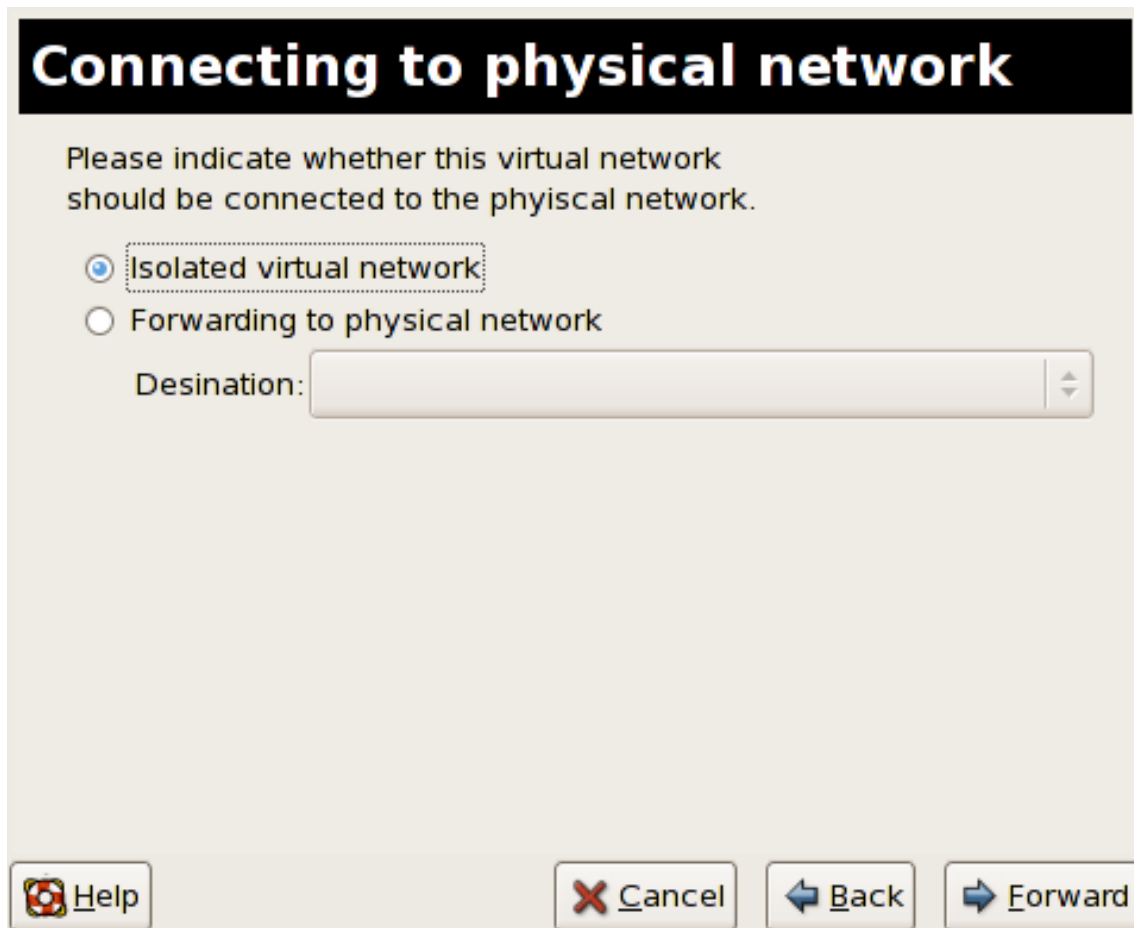


그림 17.46. Connecting to physical network

If you select **Forwarding to physical network**, choose whether the **Destination** should be **NAT to any physical device** or **NAT to physical device eth0**.

Click **Forward** to continue.

6. You are now ready to create the network. Check the configuration of your network and click **Finish**.

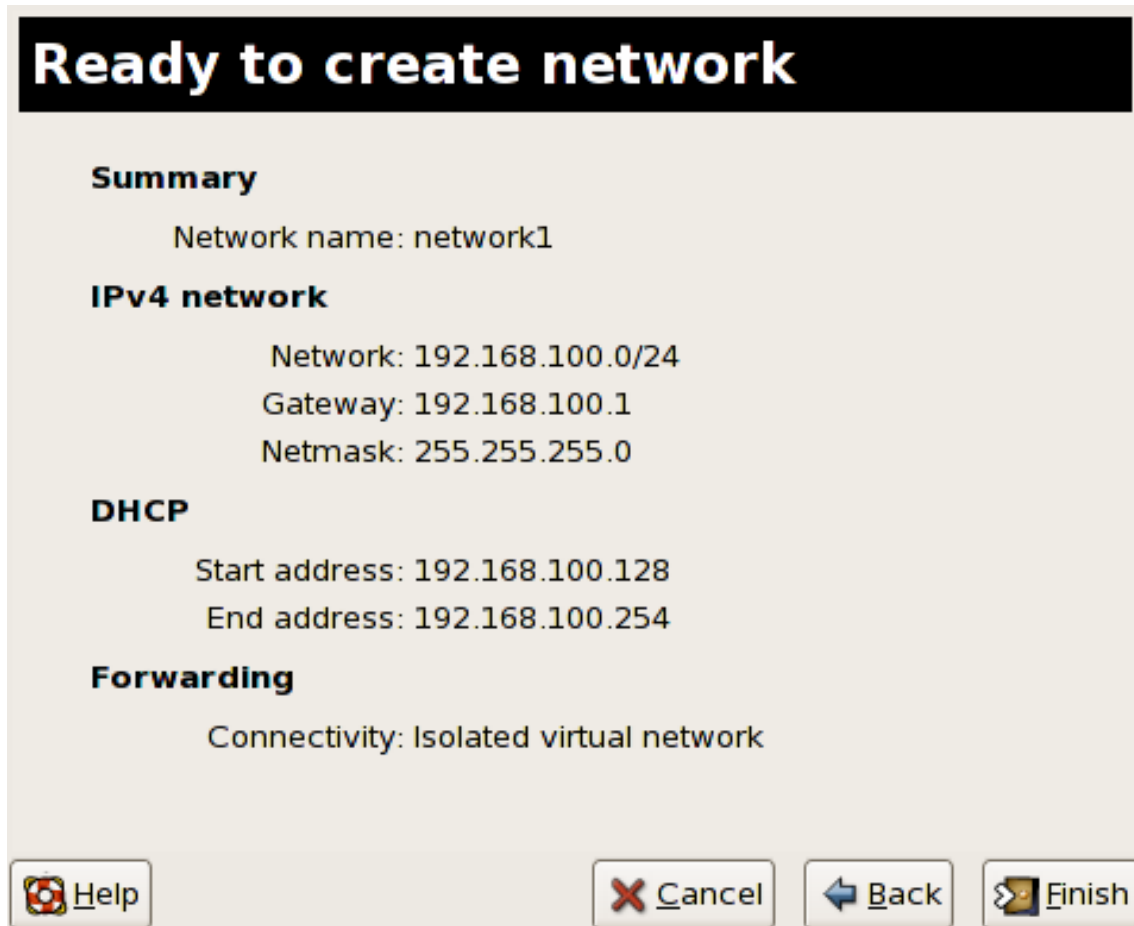


그림 17.47. Ready to create network

7. The new virtual network is now available in the **Virtual Network** tab of the **Host Details** menu.

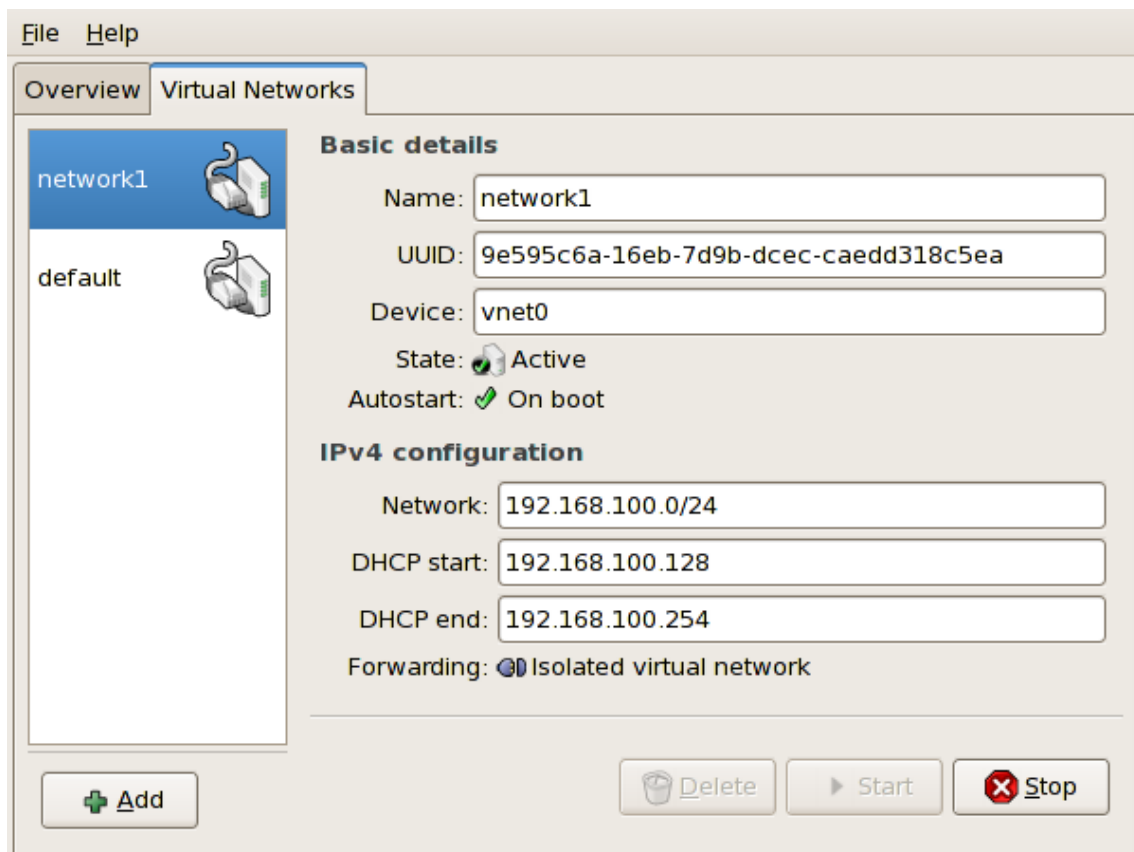


그림 17.48. New virtual network is now available

18장. RED HAT VIRTUALIZATION 문제해결

이 문제해결 단원은 사용자가 Red Hat Virtualization 시스템의 설치, 관리 또는 일반적인 일상 운영 과정에서 경험할 수 있는 문제들을 다루고 있으며, 오류 메시지, 로그 파일 위치, 시스템 도구, 데이터 검색 및 문제 분석에 대한 일반적 방법을 소개합니다.

18.1. 로그 파일 개요 및 위치

When deploying Red Hat Enterprise Linux 5.1 with Virtualization into your network infrastructure, the host's Virtualization software uses many specific directories for important configuration, log files, and other utilities. All the Red Hat Virtualization logs files are standard ASCII files, and easily accessible with any ASCII based editor:

- Red Hat Virtualization의 주요 구성 디렉토리는 **/etc/xen/**입니다. 이 디렉토리에 **xend** 데몬과 다른 가상 장치 구성 파일이 포함되어 있으며, 네트워크 스크립트 파일 또한 이 디렉토리에 위치합니다 (**/scripts** 하위 디렉토리 안).
- 문제해결 목적으로 쓰여질 모든 실제 로그 파일은 **/var/log/xen** 디렉토리에 위치합니다.
- 또한, 모든 가상 장치의 파일 기반 디스크 이미지에 사용되는 기본 디렉토리는 **/var/lib/xen** 디렉토리안에 위치하는 것을 기억하시기 바랍니다.
- **/proc** 파일 시스템에 대한 Red Hat Virtualization 정보는 **/proc/xen/** 디렉토리에 위치합니다.

18.2. 로그 파일 설명

Red Hat Virtualization에는 **xend** 데몬과 **qemu-dm** 프로세스라는 두 가지의 주요 유틸리티가 있으며, 이를 사용하여 **/var/log/xen/** 디렉토리에 다중 로그 파일을 작성합니다.

- **xend.log**는 단순한 시스템 이벤트나 운영자에 의한 작업과 같은 **xend** 데몬에 의해 수집된 모든 데이터를 포함하는 로그 파일입니다. 모든 가상 장치 운영 작업(생성, 종료, 강제 종료 등)은 여기에서 볼 수 있습니다. **xend.log**는 주로 이벤트나 수행 문제를 조사할 때 가장 먼저 살펴봐야 할 파일입니다. 이 파일은 오류 메시지에 대한 자세한 내용과 상황을 다루고 있습니다.
- **xend-debug.log**는 **xend**와 Virtualization 하위 시스템(프레임 버퍼, Python 스크립트 등)에서 이벤트 오류 기록을 다루는 로그 파일입니다.
- **xen-hotplug-log**는 핫플러그 이벤트의 데이터를 포함하는 로그 파일입니다. 장치 또는 네트워크 스크립트가 온라인으로 연결되지 않으면, 이벤트는 여기에 나타납니다.
- **qemu-dm. [PID].log**는 모든 완전 가상 게스트를 위해 **qemu-dm**에 의해 생성된 로그 파일입니다. 이 로그 파일을 사용할 때, **ps** 명령어를 사용하여 가상 장치의 **qemu-dm** 프로세스를 분리하도록 프로세스 인수를 확인하여 반드시 주어진 **qemu-dm** 프로세스 PID를 검색해야 합니다. 반드시 [PID] 기호를 실제 PID **qemu-dm** 프로세스로 대체해야 하는 것에 주의하시기 바랍니다.

가상 장치 관리자에서 오류가 발생하면, **/virt-manager** 디렉토리에 있는 **virt-manager.log** 파일에 생성된 데이터를 살펴보셔야 합니다. 매번 가상 장치 관리자를 시작할 때, 로그 파일 내용은 항상 덮어쓰여지는 것에 주의하시기 바랍니다. 따라서, 시스템 오류 발생 시 가상 장치 관리자를 재시작하기 전에 **virt-manager.log** 파일을 백업해 두시기 바랍니다.

18.3. 주요 디렉토리 위치

Red Hat Virtualization 환경에서 오류를 조사하고 문제를 해결할 때 반드시 알아두어야 할 추가적인 유틸리티와 로그 파일이 있습니다:

- 가상 장치 이미지는 `/var/lib/xen/images` 디렉토리에 위치합니다.
- `xend` 데몬을 재시작하면, `/var/lib/xen/xend-db` 디렉토리에 위치한 `xend-database`를 업데이트합니다.
- 가상 장치 덤프(`xm dump-core` 명령어에 의해 수행)는 `/var/lib/xen/dumps` 디렉토리에 위치합니다.
- `/etc/xen` 디렉토리는 시스템 자원을 관리하는 데 사용되는 구성 파일을 포함하고 있습니다. `xend` 데몬 구성 파일은 `xend-config.sxp`라고 불리며, 이 파일을 사용하여 전체 시스템 변경을 수행하고 네트워크 콜아웃을 설정합니다.
- `proc` 명령어는 시스템 정보를 검색할 수 있는 또 다른 자원입니다. 이 `proc` 항목은 `/proc/xen` 디렉토리에 위치합니다:

`/proc/xen/capabilities`

`/proc/xen/balloon`

`/proc/xen/xenbus/`

18.4. 문제해결 도구

이 단원은 시스템 관리자 프로그램, 네트워크 유틸리티, 고급 디버깅 도구에 대해 정리하여 보여줍니다 (Red Hat Virtualization 서비스의 구성 도구에 대한 더 자세한 정보는 각 구성 문서자료를 참조하시기 바랍니다). 이러한 표준 시스템 관리자 도구와 로그 파일을 이용하여 문제를 해결할 수 있습니다:

- `xentop`
- `xm dmesg`
- `xm log`
- `vmstat`
- `iostat`
- `lsof`

고급 디버깅 도구와 로그 파일을 이용하여 문제를 해결할 수 있습니다:

- `Xen0profile`
- `systemTap`
- `crash`
- `sysrq`
- `sysrq t`
- `sysrq w`

네트워크 도구를 이용해 문제를 해결할 수 있습니다:

- `ifconfig`

- `tcpdump`
- `brctl`

`brctl`은 Virtualization 리눅스 커널에서 이더넷 브리지 구성을 검사하고 설정하는 네트워크 도구입니다. 아래 예제 명령어를 실행하려면 반드시 루트 권한을 가지고 있어야 합니다:

```
# brctl show

bridge-name      bridge-id          STP  enabled  interfaces
-----
xenbr0           8000.fefffffff    no   vif13.0
xenbr1           8000.ffffeffff    yes  pddummy0
xenbr2           8000.ffffffef     no   vif0.0

# brctl showmacs xenbr0

port-no          mac-addr           local?  ageing timer
1                fe:ff:ff:ff:ff:   yes     0.00
2                fe:ff:ff:fe:ff:   yes     0.00

# brctl showstp xenbr0

xenbr0

bridge-id        8000.fefffffffffff
designated-root   8000.fefffffffffff
root-port        0                  path-cost        0
max-age          20.00             bridge-max-age   20.00
hello-time       2.00              bridge-hello-time 2.00
forward-delay    0.00              bridge-forward-delay 0.00
ageing-time      300.01
hello-timer      1.43              tcn-timer        0.00
topology-change-timer 0.00             gc-timer         0.02
```

18.5. 로그를 사용한 문제해결

When encountering issues with installing Red Hat Virtualization, you can refer to the host system's two logs to assist with troubleshooting. The `xend.log` file contains the same basic information as when you run the `xm log` command. It resides in the `/var/log/` directory. Here is an example log entry for when you create a domain running a kernel:

```
[2006-12-27 02:23:02 xend] ERROR (SrvBase: 163) op=create: Error creating domain: (0, 'Error')
```

```
Traceback (most recent call list)
File "/usr/lib/python2.4/site-packages/xen/xend/server/SrvBase.py" line 107
in perform val = op_method (op, req)
File
"/usr/lib/python2.4/site-packages/xen/xend/server/SrvDomainDir.py line 71
in op_create
raise XendError ("Error creating domain: " + str(ex))
XendError: Error creating domain: (0, 'Error')
```

다른 로그 파일인 **xend-debug.log**는 **xend.log** 보다 더 자세한 정보를 포함하고 있기에 시스템 관리자에게 매우 유용하게 쓰입니다. 아래 예제는 같은 커널 도메인 생성 시 발생하는 문제에 대한 같은 오류 데이터를 보여줍니다:

```
ERROR: Will only load images built for Xen v3.0
ERROR: Actually saw: GUEST_OS=netbsd, GUEST_VER=2.0, XEN_VER=2.0;
LOADER=generic, BSD_SYMTAB'
ERROR: Error constructing guest OS
```

고객 지원 서비스의 기술자와 연락할 때 항상 이 두 로그 파일의 복사본을 준비하시기 바랍니다.

18.6. 직렬 콘솔을 사용한 문제해결

직렬 콘솔은 어려운 문제를 해결하는 데 유용합니다. **Virtualization** 커널이 충돌하고 **hypervisor**가 오류를 생성하면, 호스트에서 오류를 조사할 방법이 전혀 없습니다. 하지만, 직렬 콘솔을 사용하면 원격 호스트에서 오류를 캡처할 수 있습니다. **Xen** 호스트를 설정하여 직렬 콘솔에 데이터를 출력하고, 원격 호스트를 설정하여 데이터를 캡처해야 합니다. 따라서, **grub.conf** 파일안의 옵션을 수정해서 **com1 /dev/ttyS0**에서 **38400-bps** 직렬 콘솔을 활성화해야 합니다.

```
title Red Hat Enterprise Linux (2.6.18-8.2080_RHEL5xen0)
  root (hd0,2)
  kernel /xen.gz-2.6.18-8.el5 com1=38400,8n1
  module /vmlinuz-2.6.18-8.el5xen ro root=LABEL=/rhgb quiet console=xvc
console=tty xencons=xvc
  module /initrd-2.6.18-8.el5xen.img
```

The **sync_console** can help determine a problem that causes hangs with asynchronous hypervisor console output, and the "**pnpci=off**" works around a problem that breaks input on the serial console. The parameters "**console=ttyS0**" and "**console=tty**" means that kernel errors get logged with on both the normal VGA console and on the serial console. Then you can install and set up **ttywatch** to capture the data on a remote host connected by a standard null-modem cable. For example, on the remote host you could type:

```
ttywatch --name myhost --port /dev/ttyS0
```

/dev/ttyS0의 출력을 **/var/log/ttywatch/myhost.log** 파일로 보냅니다.

18.7. 반 가상화 게스트 콘솔 사용

반 가상화 게스트 운영 체제는 기본적으로 데이터를 **domain0** 운영 체제로 보내도록 설정된 가상 텍스트 콘솔을 가지고 있습니다. 명령행에서 다음을 입력하여 가상 텍스트 콘솔을 사용할 수 있습니다:

```
xm console [domain name or number]
```

여기서 **domain100**은 실행 중인 이름과 번호를 의미합니다. 또한, 가상 장치 관리자를 사용하여 가상 텍스트 콘솔을 표시할 수 있습니다. 가상 장치 상세정보 창의 **보기** 메뉴에서 **직렬 콘솔**을 선택하시기 바랍니다.

18.8. 완전 가상화 게스트 콘솔 사용

Full Virtualized guest operating systems automatically has a text console configured for use, but the difference is the kernel guest is not configured. To enable the guest virtual serial console to work with the Full Virtualized guest, you must modify the guest's **grub.conf** file, and include the '**console =ttyS0 console=tty0**' parameter. This ensures that the kernel messages are sent to the virtual serial console (and the normal graphical console). If you plan to use the virtual serial console in a full virtualized guest, you must edit the configuration file in the **/etc/xen/** directory. On the host domain, you can then access the text console by typing:

```
xm console
```

또한, 가상 장치 관리자를 사용하여 직렬 콘솔을 표시할 수 있습니다. 가상 장치 상세정보 창에서, 보기 메뉴에서 직렬 콘솔을 선택하시기 바랍니다.

18.9. LUN 지속성 구현

시스템이 다중 경로를 지원하지 않으면, **udev**를 사용하여 **lun** 지속성을 구현할 수 있습니다. 시스템에 **lun** 지속성을 구현하기 전에 적절한 **UUID**를 갖고 있는지 확인하시기 바랍니다. **UUID**를 갖고 있으면 **/etc** 디렉토리의 **scsi_id** 파일을 편집하여 **lun** 지속성을 구현할 수 있습니다. 편집 도구를 사용해 파일을 열어서 다음 줄의 주석처리를 해제해야 합니다.

```
# options=-b
```

그 다음에 이 파라미터로 대체합니다:

```
# options=-g
```

이것은 **udev**가 모든 시스템 **SCSI** 장치를 검사하여 **UUID**를 가져옵니다. 시스템 **UUID**를 확인하려면 다음을 입력하시기 바랍니다:

```
# scsi_id -g -s /block/sdc
```

결과는 다음과 비슷합니다:

```
[root@devices] # scsi_id -g -s /block/sdc
*3600a0b80001327510000015427b625e*
```

이 긴 문자열이 **UUID**입니다. **UUID**를 이용해 장치 이름을 확인하려면, 각 장치 경로를 검사하여 **UUID** 번호가 각 장치와 같은지 확인합니다. 시스템에 새로운 장치를 추가할 때 **UUID**는 변하지 않습니다. 장치 경로를 검사하면 장치 이름에 대한 규칙을 반드시 만들어야 합니다. 이러한 규칙을 생성하려면 **/etc/udev/rules.d** 디렉토리에 위치한 **20-names.rules** 파일을 편집합니다. 여기서 생성한 장치 이름 규칙은 다음 형식을 따릅니다.

```
# KERNEL="sd*", BUS="scsi", PROGRAM="sbin/scsi_id", RESULT="UUID",
NAME="devicename"
```

현재 사용 중인 UUID와 장치 이름을 위의 UUID 검색 항목으로 대체하시기 바랍니다. 규칙은 다음과 비슷합니다:

```
KERNEL="sd*", BUS="scsi", PROGRAM="sbin/scsi_id",
RESULT="3600a0b80001327510000015427b625e
", NAME="mydevicename"
```

이것은 시스템이 주어진 UUID를 검사하고자 `/dev/sd*`와 일치하는 모든 장치를 활성화하도록 해줍니다. 일치된 장치를 검색했을 때, `/dev/devicename`이라는 장치 노드를 생성합니다. 이 예제에서 장치 노드는 `/dev/mydevice`입니다. 마지막으로, `/etc` 디렉토리에 위치한 `rc.local` 파일에 다음 경로를 추가해야 합니다:

```
/sbin/start_udev
```

다중 경로를 사용한 lun 지속성 구현

다중 경로 환경에서 lun 지속성을 구현하려면 다중 경로 장치에 대한 별칭을 정의해야 합니다. 이 예제에는 `/etc/` 디렉토리의 `multipath.conf` 파일을 편집하여 네 가지 장치를 정의합니다:

```
multipath {
    wwid      3600a0b80001327510000015427b625e
    alias     oramp1
}
multipath {
    wwid      3600a0b80001327510000015427b6
    alias     oramp2
}
multipath {
    wwid      3600a0b80001327510000015427b625e
    alias     oramp3
}
multipath {
    wwid      3600a0b80001327510000015427b625e
    alias     oramp4
}
```

네 종류의 lun을 정의합니다: `/dev/mpath/oramp1`, `/dev/mpath/oramp2`, `/dev/mpath/oramp3`, `dev/mpath/oramp4`. 이 장치들은 `/dev/mpath` 디렉토리에 위치하며 lun 이름은 lun의 wwid에서 생성된 별칭에 의해 재부팅과 관계없이 지속적으로 사용됩니다.

18.10. SELINUX 고려사항

이 단원은 Red Hat Virtualization 환경에 SELinux를 수행할 때 반드시 고려해야 하는 사항들을 다루고 있습니다. 시스템을 변경하거나 장치를 추가할 때, 해당하는 SELinux 정책을 반드시 업데이트해야 합니다. 게스트에 대한 LVM을 구성하려면, 각 기초 블록 장치와 볼륨 그룹에 대한 SELinux 문맥을 반드시 수정해야 합니다.

```
# semanage fcontext -a -t xen_image _t -f -b /dev/sda2
# restorecon /dev/sda2
```

불 파라미터인 `xend_disable_trans`는 데몬이 재시작한 후에 `xend`를 자유모드에 설정합니다. 전체 시스템보다 하나의 데몬에 대한 보호 기능을 비활성화하는 것이 좋습니다. 또한, 다른 목적으로 사용할 수 있는 `xen_image_t`로서 디렉토리를 다시 레이블하지 않기를 권장합니다.

18.11. 게스트 디스크 이미지에서 데이터 사용

두 가지의 다른 프로그램을 사용하여 게스트 디스크 이미지로부터 데이터를 사용할 수 있습니다. 이 프로그램을 사용하기 전에 반드시 게스트를 종료해야 합니다. 게스트와 `dom0`의 파일 시스템에 접근하는 것은 시스템에 위험을 가져올 수도 있습니다.

`kpartx` 프로그램을 사용하여 파티션된 디스크 또는 LVM 볼륨 그룹을 조작할 수 있습니다:

```
yum install kpartx
kpartx -av /dev/xen/guest1
add map guest1p1 : 0 208782 linear /dev/xen/guest1 63
add map guest1p2: 0 16563015 linear /dev/xen/guest1 208845
```

두 번째 파티션에서 LVM 볼륨을 사용하려면, 반드시 `vgscan`을 사용하여 LVM을 다시 검사하고 `vgchange -ay` 명령어를 사용하여 파티션(디폴트로 `VolGroup00`라고 볼륨)에서 볼륨 그룹을 활성화해야 합니다:

```
# kpartx -a /dev/xen/guest1
#vgscan
Reading all physical volumes . This may take a while...
Found volume group "VolGroup00" using metadata type lvm2
# vgchange -ay VolGroup00
2 logical volume(s) in volume group VolGroup00 now active.
# lvs
LV VG Attr Lsize Origin Snap% Move Log Copy%
LogVol00 VolGroup00 -wi-a- 5.06G
LogVol01 VolGroup00 -wi-a- 800.00M
# mount /dev/VolGroup00/LogVol00 /mnt/
....
#umount /mnt/
#vgchange -an VolGroup00
#kpartx -d /dev/xen/guest1
```

모두 완료되면, `vgchange -an`을 사용하여 논리 볼륨을 비활성화하고, `kpartx -d`를 사용하여 파티션을 제거하고, `losetup -d` 를 사용하여 루프 장치를 제거해야 하는 것을 반드시 기억하시기 바랍니다.

18.12. 일반 문제해결 상황

`xend` 서비스를 실행할 때 아무것도 발생하지 않습니다. `xm list1`을 입력하면 다음 내용이 출력될 것입니다:

```
Error: Error connecting to xend: Connection refused. Is xend running?
```

`xend start`를 수동으로 실행하면 다음과 같이 더 많은 오류가 뜰 것입니다:

```
Error: Could not obtain handle on privileged command interfaces (2 = No
such file or directory)
Traceback (most recent call last:)
```

```
File "/usr/sbin/xend/", line 33 in ?
```

```

from xen.xend.server import SrvDaemon

File "/usr/lib/python2.4/site-packages/xen/xend/server/SrvDaemon.py" ,
line 26 in ?

from xen.xend import XendDomain

File "/usr//lib/python2.4/site-packages/xen/xend/XendDomain.py" , line 33,
in ?

from xen.xend import XendDomainInfo

File "/usr/lib/python2.4/site-packages/xen/xend/image.py" , line37, in ?

import images

File "/usr/lib/python2.4/site-packages/xen/xend/image.py" , line30, in ?

xc = xen.lowlevel.xc.xc ()

RuntimeError: (2, 'No such file or directory' )

```

위 결과는 **xen-hypervisor**가 아닌 커널로 호스트를 재부팅했을 때 발생할 수 있는 상황입니다. 이 문제를 해결하려면, 시작 시 반드시 **xen-hypervisor** 커널을 선택하거나 **grub.conf** 파일에 **xen-hypervisor** 커널을 디폴트로 설정해야 합니다.

18.13. 루프 장치 오류

파일 기반 게스트 이미지를 사용할 때, 설정된 루프 장치 개수를 늘릴 필요가 있을 것입니다 (디폴트로 최대 8개의 루프 장치를 활성화할 수 있습니다). 8개 보다 많은 파일 기반 게스트/루프 장치가 필요하다면, **/etc/modprobe.conf** 파일을 수정하며, **modprobe.conf** 파일 안에 반드시 다음을 써넣어야 합니다.

```
options loop max_loop=64
```

이 예제에서 최대 루프 값으로 64를 사용하지만, 다른 숫자를 지정할 수도 있습니다. 또한, 시스템에서 루프 장치 기반 게스트를 실행해야 합니다. 루프 장치 기반 게스트를 반 가상 시스템에 도입하려면 **phy: block device** 또는 **tap:aio** 명령어를 사용하시기 바랍니다. 완전 가상 시스템에 대한 루프 장치 기반 게스트는 **phy: device** 또는 **file: file** 명령어를 사용합니다.

18.14. 게스트 생성 오류

When you attempt to create a guest, you receive an **"Invalid argument"** error message. This usually means that the kernel image you are trying to boot is incompatible with the hypervisor. An example of this would be if you were attempting to run a non-PAE FC5 kernel on a PAE only FC6 hypervisor.

yum 업데이트를 통해 새로운 커널을 받고, **grub.conf**에서 기본 커널로 **Virtualization** 커널 대신 완전한 커널로 전환합니다.

이 문제를 해결하려면, **/etc/sysconfig/kernel/** 디렉토리에 있는 기본 커널 RPM을 수정해야 합니다. 또한, **kernel-xen** 파라미터가 **gb.conf** 파일에 기본 옵션으로 설정되어 있는지 확인하시기 바랍니다.

18.15. 직렬 콘솔 오류

직렬 콘솔에 아무런 출력이 없습니다. 이 문제를 해결하려면, **grub.conf** 파일을 수정하여 com 포트 파라미터를 다음과 같이 변경해야 합니다:

```
serial --unit=1 --speed=115200

title RHEL5 i386 Xen (2.6.18-1.2910.el5xen)
root (hd0, 8)
kernel /boot/xen.gz-2.6.18-1.2910.el5 com2=115200,8n1
module /boot/vmlinuz-2.6.18-1.2910.el5xen to root=LABEL=RHEL5_i386
console=tty console=ttyS1115200
module /boot/initrd-2.8.6.18-12910.el5xen.img

title RHEL5 i386 xen (2.6.18.-1.2910.el5xen)
root (hd0, 8)
kernel /boot/xen.gz-2.6.18-1.2910.el5 com2=115200 console=com21
module /boot/vmlinuz2.6.18-1.2910.el5xen to root=LABEL=RHEL5_i386
console=xvc xencons=xvc
module /boot/ititrd-2.6.18-1.2910.el5xen.img
```

grub.conf 파일을 위와 같이 변경하면 직렬 콘솔이 제대로 작동할 것입니다. **ttys**에는 어떤 수도 사용할 수 있으며, **ttys0**와 같이 작동되어야 합니다.

18.16. 네트워크 브리지 오류

Red Hat Virtualization can configure multiple Virtualization network bridges to use with multiple ethernet cards. To successfully configure multiple network bridges for ethernet cards, you must configure the second network interface by either using the system-config-network TUI/GUI, or by creating a new configuration file in **/etc/sysconfig/network-scripts** . You should use a process to setup multiple Xen bridges. This is an example config file for a second NIC called '**eth1**' :

```
#/etc/sysconfig/network-scripts/fcfg-eth1
DEVICE=eth1
BOOTPROTO=static
ONBOOT=yes
USERCTL=no
IPV6INIT=no
PEERDNS=yes
TYPE=Ethernet
NETMASK=255.255.255.0
IPADDR=10.1.1.1
GATEWAY=10.1.1.254
ARP=yes
```

/etc/xen/scripts/network-bridge 파일을 **/etc/xen/scripts/network-bridge.xen** 파일로 복사하시기 바랍니다.

Edit **/etc/xen/xend-config.sxp** and add a line to your new network bridge script (this example uses "**network-virtualization-multi-bridge**").

xend-config.sxp 파일에서 새로운 줄은 새 스크립트를 반영해야 합니다:

```
network-script network-xen-multi-bridge
```

다음은 나타내는 줄에 주석처리를 해제하시기 바랍니다:

```
network-script network-bridge
```

다중 Xen 브리지를 생성하려면, 사용자 환경에 맞는 스크립트를 생성해야 합니다. 아래 예제는 **xenbr0**와 **xenbr1**이라는 두 개의 Xen 브리지를 생성하고 **eth1**과 **eth0**에 각각 덧붙입니다:

```
# !/bin/sh
# network-xen-multi-bridge
# Exit if anything goes wrong
set -e
# First arg is operation.
OP=$1
shift
script=/etc/xen/scripts/network-bridge.xen
case ${OP} in
start)
$script start vifnum=1 bridge=xenbr1 netdev=eth1
$script start vifnum=0 bridge=xenbr0 netdev=eth0
;;
stop)
$script stop vifnum=1 bridge=xenbr1 netdev=eth1
$script stop vifnum=0 bridge=xenbr0 netdev=eth0
;;
status)
$script status vifnum=1 bridge=xenbr1 netdev=eth1
$script status vifnum=0 bridge=xenbr0 netdev=eth0
;;
*)
echo 'Unknown command: ' ${OP}
echo 'Valid commands are: start, stop, status'
exit 1
esac
```

추가 브리지를 생성하려면, 간단히 예제 스크립트를 사용하여 파일을 복사/붙여넣기 하시기 바랍니다.

18.17. 랩톱 구성

The task of configuring your RHEL 5.1 loaded laptop for use on a network environment, presents a number of potential challenges. Most WiFi and wired connections switch constantly during any given day, and Red Hat Virtualization assumes it has access to the same interface consistently. This results in the system performing ifup/ifdown calls to the network interface in use by Red Hat Virtualization. WiFi cards are not the ideal network connection method since Red Hat Virtualization uses the default network interface.

The idea here is to create a 'dummy' network interface for Red Hat Virtualization to use.

This technique allows you to use a hidden IP address space for your guests and Virtual Machines. To do this operation successfully, you must use static IP addresses as DHCP does not listen for IP addresses on the dummy network. You also must configure NAT/IP masquerading to enable network access for your guests and Virtual Machines. You should attach a static IP when you create the 'dummy' network interface.

다음 예제에서 인터페이스를 `dummy0`라고 부르며 IP 주소로 `10.1.1.1`을 사용합니다. 이 스크립트는 `ifcfg-dummy0`이며, `/etc/sysconfig/network-scripts/` 디렉토리에 위치합니다:

```
DEVICE=dummy0
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
IPV6INIT=no
PEERDNS=yes
TYPE=Ethernet
NETMASK=255.255.255.0
IPADDR=10.1.1.1
ARP=yes
```

`xenbr0`를 `dummy0`로 바인드하여 실제 네트워크로부터 차단될 때도 네트워크 연결을 허용해야 합니다.

You will need to make additional modifications to the `xend-config.sxp` file. You must locate the (`network-script 'network-bridge' bridge=xenbr0`) section and add include this in the end of the line:

```
netdev=dummy0
```

You must also make some modifications to your guest's domU networking configuration to enable the default gateway to point to `dummy0`. You must edit the DomU 'network' file that resides in the `/etc/sysconfig/` directory to reflect the example below:

```
NETWORKING=yes
HOSTNAME=localhost.localdomain
GATEWAY=10.1.1.1

IPADDR=10.1.1.10
NETMASK=255.255.255.0
```

domU가 공개 네트워크를 사용하도록 NAT를 활성화하는 것은 바람직합니다. NAT를 사용하면 무선 네트워크 사용자도 Red Hat Virtualization 무선 제한없이 네트워크를 사용할 수 있습니다. `/etc/rc3.d` 디렉토리안의 `S99XenLaptopNAT` 파일을 아래와 같이 수정하시기 바랍니다:

```
#!/bin/bash
#
# XenLaptopNAT Startup script for Xen on Laptops
#
# chkconfig: - 99 01
# description: Start NAT for Xen Laptops
#
# PATH=/usr/bin:/sbin:/bin:/usr/sbin
# export PATH
GATEWAYDEV=`ip route | grep default | awk {'print $5'}`
iptables -F
case "$1" in
start)
if test -z "$GATEWAYDEV"; then
echo "No gateway device found"
else
echo "Masquerading using $GATEWAYDEV"
```

```

/sbin/iptables -t nat -A POSTROUTING -o $GATEWAYDEV -j MASQUERADE
fi
echo "Enabling IP forwarding"
echo 1 > /proc/sys/net/ipv4/ip_forward
echo "IP forwarding set to `cat /proc/sys/net/ipv4/ip_forward`"
echo "done."
;;
*)
echo "Usage: $0 {start|restart|status}"
;;
esac

```

시작 시 자동으로 네트워크 설정이 필요하면, `/etc/rc3.d/S99XenLaptopNAT`에 대한 소프트 링크를 생성해야 합니다.

`modprobe.conf` 파일을 수정할 때 다음 줄을 써넣어야 합니다:

```

alias dummy0 dummy
options dummy numdummies=1

```

18.18. STARTING DOMAINS AUTOMATICALLY DURING SYSTEM BOOT

Starting Domains Automatically During System Boot

시스템이 시작할 때 자동으로 게스트를 시작하도록 설정할 수 있습니다. 이것은 `/etc/xen/auto`에 있는 심볼릭 링크를 수정하여 이루어지며, 이 심볼릭 링크 파일은 자동으로 시작되는 게스트 구성 파일을 가리킵니다. 시작 과정은 게스트 번호가 높을수록 부트 과정은 더 길어지도록 나열됩니다. 아래 예제는 게스트 `rhe15vm01`에 대해 어떻게 심볼릭 링크를 사용하는지 보여줍니다:

```

[root@python xen]# cd /etc/xen
[root@python xen]# cd auto
[root@python auto]# ls
[root@python auto]# ln -s ../rhe15vm01 .
[root@python auto]# ls -l

lrwxrwxrwx 1 root root 14 Dec 14 10:02 rhe15vm01 -> ../rhe15vm01

[root@python auto]#

```

18.19. DOMAIN0 수정

To use Red Hat Virtualization to manage domain0, you will constantly making changes to the `grub.conf` configuration file, that resides in the `/etc` directory. Because of the large number of domains to manage, many system administrators prefer to use the 'cut and paste' method when editing `grub.conf`. If you do this, make sure that you include all five lines in the Virtualization entry (or this will create system errors). If you require Xen hypervisor specific values, you must add them to the 'xen' line. This example represents a correct `grub.conf` Virtualization entry:

```

# boot=/dev/sda/
default=0
timeout=15
#splashimage=(hd0, 0)/grub/splash.xpm.gz

```

```

hiddenmenu
serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
terminal --timeout=10 serial console
title Red Hat Enterprise Linux Server (2.6.17-1.2519.4.21. el5xen)
    root (hd0, 0)
    kernel /xen.gz-2.6.17-1.2519.4.21.el5 com1=115200,8n1
    module /vmlinuz-2.6.17-1.2519.4.21el5xen ro root=/dev/VolGroup00/LogVol100
    module /initrd-2.6.17-1.2519.4.21.el5xen.img

```

For example, if you need to change your dom0 hypervisor's memory to 256MB at boot time, you must edit the 'xen' line and append it with the correct entry, '**dom0_mem=256M**'. This example represents the respective **grub.conf** xen entry:

```

# boot=/dev/sda
default=0
timeout=15
#splashimage=(hd0,0)/grubs/splash.xpm.gz
hiddenmenu
serial --unit=0 --speed =115200 --word=8 --parity=no --stop=1
terminal --timeout=10 serial console
title Red Hat Enterprise Linux Server (2.6.17-1.2519.4.21. el5xen)
    root (hd0,0)
    kernel /xen.gz-2.6.17-1.2519.4.21.el5 com1=115200,8n1 dom0_mem=256MB
    module /vmlinuz-2.6.17-1.2519.4.21.el5xen ro
    root=/dev/VolGroup00/LogVol100
    module /initrd-2.6.17-1.2519.4.21.el5xen.img

```

18.20. 게스트 구성 파일

When you install new guests using virt-manager (or virt-install) tool(s) from Red Hat Enterprise Linux 5.1 with Virtualization, the guests configuration files (located in the /etc/xen directory) get modified and setup automatically. This configuration file example is for a para-virtualized guest:

```

name = "rhel5vm01"
memory = "2048"
disk = ['tap:aio:/xen/images/rhel5vm01.dsk,xvda,w',]
vif = ["type=ieomu, mac=00:16:3e:09:f0:12 bridge=xenbr0",
      "type=ieomu, mac=00:16:3e:09:f0:13 ]
vnc = 1
vncunused = 1
uuid = "302bd9ce-4f60-fc67-9e40-7a77d9b4e1ed"
bootloader = "/usr/bin/pygrub"
vcpus=2
on_reboot = "restart"
on_crash = "restart"

```

Note that the **serial="pty"** is the default for the configuration file. This configuration file example is for a fully-virtualized guest:

```

name = "rhel5u5-86_64"
builder = "hvm"
memory = 500
disk = ['file:/xen/images/rhel5u5-x86_64.dsk.hda,w']
vif = [ 'type=ioemu, mac=00:16:3e:09:f0:12, bridge=xenbr0', 'type=ieomu,

```

```

mac=00:16:3e:09:f0:13, bridge=xenbr1']
uuid = "b10372f9-91d7-ao5f-12ff-372100c99af5'
device_model = "/usr/lib64/xen/bin/qemu-dm"
kernel = "/usr/lib/xen/boot/hvmlloader/"
vnc = 1
vncunused = 1
apic = 1
acpi = 1
pae = 1
vcpus =1
serial ="pty" # enable serial console
on_boot = 'restart'

```

18.21. 게스트 구성 파일 클론

You can copy (or clone) an existing configuration file to create an all new guest. You must modify the name parameter of the guests' configuration file. The new, unique name then appears in the hypervisor and is viewable by the management utilities. You must generate an all new UUID as well (using the **uuidgen(1)** command). Then for the **vif** entries you must define a unique MAC address for each guest (if you are copying a guest configuration from an existing guest, you can create a script to handle it). For the xen bridge information, if you move an existing guest configuration file to a new host, you must update the **xenbr** entry to match your local networking configuration. For the Device entries, you must modify the entries in the '**disk='** section to point to the correct guest image.

You must also modify these system configuration settings on your guest. You must modify the HOSTNAME entry of the **/etc/sysconfig/network** file to match the new guest's hostname.

/etc/sysconfig/network-scripts/ifcfg-eth0 파일에 있는 HWADDR 주소를 수정하여 **ifconfig eth0** 출력과 일치시키고, 수동 IP 주소를 사용하려면 IPADDR 항목을 수정해야 합니다.

18.22. MAC 주소를 생성하는 스크립트 구성

Red Hat Virtualization은 가상 장치가 생성될 때 각 가상 장치에 대한 MAC 주소를 생성할 수 있습니다. 같은 서브넷에 거의 무제한의 MAC 주소 개수가 존재하기에 같은 MAC 주소를 갖는 경우는 발생하지 않을 것입니다. 또한 MAC 주소를 생성할 수 있는 스크립트를 구성하여 이 문제를 해결할 수도 있습니다. 아래 예제는 MAC 주소 생성 파라미터를 포함한 스크립트를 보여줍니다:

```

#!/usr/bin/python
# macgen.py script generates a MAC address for Xen guests
#
import random
mac = [ 0x00, 0x16, 0x3e,
        random.randint(0x00, 0x7f),
        random.randint(0x00, 0xff),
        random.randint(0x00, 0xff) ]
print ':'.join(map(lambda x: "%02x" % x, mac))

Generates e.g.:
00:16:3e:66:f5:77
to stdout

```

18.23. 가상 장치 실시간 이전 설정

Red Hat Virtualization can migrate virtual machines between other servers running Red Hat Enterprise Linux 5.1 with Virtualization. Further, migration is performed in an offline method (using the **xm migrate** command). Live migration can be done from the same command. However there are some additional modifications that you must do to the **xend-config** configuration file. This example identifies the entries that you must modify to ensure a successful migration:

(xend-relocation-server yes)

The default for this parameter is 'no', which keeps the relocation/migration server deactivated (unless on a trusted network) and the domain virtual memory is exchanged in raw form without encryption.

(xend-relocation-port 8002)

이 파라미터는 **xend**가 이전에 사용할 포트를 설정합니다. 값을 정확히 입력하신 후 앞의 주석 처리를 해제하시기 바랍니다.

(xend-relocation-address)

이 파라미터는 **xend-relocation-server**를 활성화한 후에 소켓 연결 재배치를 위해 대기하는 주소를 가리킵니다. 대기 중에 특정 인터페이스에 대한 이동을 제한합니다.

(xend-relocation-hosts-allow)

This parameter controls the host that communicates with the relocation port. If the value is empty, then all incoming connections are allowed. You must change this to a space-separated sequences of regular expressions (such as **xend-relocation-hosts-allow-'^localhost\\.localdomain\$'**). A host with a fully qualified domain name or IP address that matches these expressions are accepted.

위 파라미터를 설정한 후에 Red Hat Virtualization 호스트를 재시작하여 새 파라미터를 적용하시기 바랍니다.

18.24. 오류 메시지 해석

다음 오류 메시지를 볼 수 있습니다:

```
failed domain creation due to memory shortage, unable to balloon domain0
```

충분한 RAM 공간이 없으므로 도메인이 실패했습니다. domain0에 새롭게 생성되는 게스트에 필요한 메모리 공간을 제공하도록 확장할 수 없습니다. 이 오류에 대해 **xend.log** 파일을 확인해 보시기 바랍니다:

```
[2006-12-21] 20:33:31 xend 3198] DEBUG (balloon:133) Balloon: 558432 Kib
free; 0 to scrub; need 1048576; retries: 20
[2006-12-21] 20:33:31 xend. XendDomainInfo 3198] ERROR (XendDomainInfo:
202
Domain construction failed
```

You can check the amount of memory in use by domain0 by using the **xm list Domain0** command. If domain0 is not ballooned down, you can use the command "**xm mem-set Domain-0 NewMemSize**" to check memory.

다음 오류 메시지를 볼 수 있습니다:

```
wrong kernel image: non-PAE kernel on a PAE
```

This message indicates that you are trying to run an unsupported guest kernel image on your Hypervisor. This happens when you try to boot a non-PAE paravirtual guest kernel on a RHEL 5.1 hypervisor. Red Hat Virtualization only supports guest kernels with PAE and 64bit architectures.

다음 명령어를 입력하시기 바랍니다:

```
[root@smith]# xm create -c va base

Using config file "va-base"
Error: (22, 'invalid argument')
[2006-12-14 14:55:46 xend.XendDomainInfo 3874] ERRORs
(XendDomainInfo:202) Domain construction failed

Traceback (most recent call last)
File "/usr/lib/python2.4/site-packages/xen/xend/XendDomainInfo.py", line
195 in create vm.initDomain()
File " /usr/lib/python2.4/site-packages/xen/xend/XendDomainInfo.py", line
1363 in initDomain raise VmError(str(exn))
VmError: (22, 'Invalid argument')
[2006-12-14 14:55:46 xend.XendDomainInfo 3874] DEBUG (XenDomainInfo: 1449]
XendDomainInfo.destroy: domin=1
[2006-12-14 14:55:46 xend.XendDomainInfo 3874] DEBUG (XenDomainInfo: 1457]
XendDomainInfo.destroy:Domain(1)
```

32비트이며 PAE가 아닌 커널을 실행하려면 게스트를 완전 가상화된 가상 장치로서 실행해야 합니다. 반 가상화 게스트에 32비트 PAE 게스트를 실행하고자 한다면, 반드시 32비트 PAE hypervisor를 가지고 있어야 하며, 64비트 PAE 게스트를 실행하려면 64비트 PAE hypervisor를 가지고 있어야 합니다. 완전 가상화 게스트에는 64비트 hypervisor에서 64비트 게스트를 실행해야 합니다. RHEL 5 i686에서 32비트 PAE hypervisor는 단지 32비트 PAE 반 가상화와 32비트 완전 가상화 게스트 OS만을 지원하며, 64비트 hypervisor는 64비트 반 가상화 게스트만을 지원합니다.

This happens when you move the full virtualized HVM guest onto a RHEL 5.1 system. Your guest may fail to boot and you will see an error in the console screen. Check the PAE entry in your configuration file and ensure that `paе=1`. You should use a 32bit distribution.

다음 오류 메시지를 볼 수 있습니다:

```
Unable to open a connection to the Xen hypervisor or daemon
```

이 오류 메시지는 가상 장치 관리자 프로그램이 시작하지 못하고 실패하였을 때 볼 수 있으며, 주로 `/etc/hosts` 구성 파일에 `localhost` 항목이 없을 때 발생합니다. 파일을 확인하여 `localhost` 항목이 활성화되어 있는지 확인하시기 바랍니다. 아래 예제는 부정확한 `localhost` 항목을 보여줍니다:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
localhost.localdomain localhost
```

아래 예제는 정확한 `localhost` 항목을 보여줍니다:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 localhost.localdomain localhost
localhost.localdomain. localhost
```

`xen-xend.log` file 파일에서 다음과 같은 오류를 볼 수 있습니다:


```
Bridge xenbr1 does not exist!
```

This happens when the guest's bridge is incorrectly configured and this forces the Xen hotplug scripts to timeout. If you move configuration files between hosts, you must ensure that you update the guest configuration files to reflect network topology and configuration modifications. When you attempt to start a guest that has an incorrect or non-existent Xen bridge configuration, you will receive the following errors:

```
[root@trumble virt]# xm create r5b2-mysQL01

Using config file " r5b2-mysQL01"
Going to boot Red Hat Enterprise Linux Server (2.6.18.-1.2747 .e15xen)
kernel: /vmlinuz-2.6.18-12747.e15xen
initrd: /initrd-2.6.18-1.2747.e15xen.img
Error: Device 0 (vif) could not be connected. Hotplug scripts not working.
```

또한, **xend.log**는 다음과 같은 오류를 표시합니다:

```
[2006-11-14 15:07:08 xend 3875] DEBUG (DevController:143) Waiting for
devices vif
[2006-11-14 15:07:08 xend 3875] DEBUG (DevController:149) Waiting for 0
[2006-11-14 15:07:08 xend 3875] DEBUG (DevController:464)
hotplugStatusCallback

/local/domain/0/backend/vif/2/0/hotplug-status

[2006-11-14 15:08:09 xend.XendDomainInfo 3875] DEBUG (XendDomainInfo:1449)
XendDomainInfo.destroy: domid=2
[2006-11-14 15:08:09 xend.XendDomainInfo 3875] DEBUG (XendDomainInfo:1457)
XendDomainInfo.destroyDomain(2)
[2006-11-14 15:07:08 xend 3875] DEBUG (DevController:464)
hotplugStatusCallback

/local/domain/0/backend/vif/2/0/hotplug-status
```

이 문제를 해결하려면 게스트 구성 파일의 **vif** 항목을 수정하여 편집해야 합니다. 디폴트 브리지로 **xenbr0**를 사용하고 있다고 가정하면, 구성 파일의 **vif** 항목은 다음과 비슷할 것입니다:

```
# vif = ['mac=00:16:3e:49:1d:11, bridge=xenbr0',]
```

다음과 같은 python deprecation 오류를 볼 수 있습니다:

```
[root@python xen]# xm shutdown win2k3xen12
[root@python xen]# xm create win2k3xen12

Using config file "win2k3xen12".

/usr/lib64/python2.4/site-packages/xenxm/opts.py:520: Deprecation Warning:
Non ASCII character '\xc0' in file win2k3xen12 on line 1, but no encoding
declared; see http://www.python.org/peps/pep-0263.html for details

execfile (defconfig, globs, locs,)
Error: invalid syntax 9win2k3xen12, line1)
```

Python은 유효하지 않은(또는 부정확한) 구성파일에서 이 오류 메시지를 생성합니다. 이 문제를 해결하려면 부정확한 구성 파일을 수정하거나 새로운 구성 파일을 생성해야 합니다.

18.25. 온라인 문제해결 자료

- Red Hat Virtualization 센터

<http://www.openvirtualization.com>

- Red Hat Enterprise Linux 5 Beta 2 문서자료

<http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/index.html>

- Libvirt API

<http://www.libvirt.org>

- 가상 장치 관리자 프로젝트 홈 페이지

<http://virt-manager.et.redhat.com>

- Xen 커뮤니티 센터

<http://www.xensource.com/xen/xen/>

- 가상화 기술 개요

<http://virt.kernelnewbies.org>

- 통합 기술 프로젝트

<http://et.redhat.com>

19장. 추가 자료

Red Hat Virtualization에 관한 더 자세한 정보를 얻으려면, 다음 자료를 참조하시기 바랍니다.

19.1. 유용한 웹사이트

- <http://www.cl.cam.ac.uk/research/srg/netos/xen/> — Red Hat Virtualization에 따른 Xen™ 반 가상화 장치 관리자의 프로젝트 웹사이트입니다. 이 사이트는 업스트림 Xen 프로젝트 바이너리와 소스 코드를 유지하며, Xen 관련 기술은 물론 Xen에 관한 정보, 구조 개요, 문서 자료, 관련 링크도 다루고 있습니다.
- <http://www.libvirt.org/> — 호스트 OS의 가상화 프레임워크와 연동하는 **libvirt** 가상화 API의 공식 웹사이트입니다.
- <http://virt-manager.et.redhat.com/> — 가상 장치를 관리하는 그래픽형식 프로그램인 가상 장치 관리자의 프로젝트 웹사이트입니다.

19.2. 설치된 문서 자료

- `/usr/share/doc/xen-<version-number>/` — This directory contains a wealth of information about the Xen para-virtualization hypervisor and associated management tools, including a look at various example configurations, hardware-specific information, and the current Xen upstream user documentation.
- `man virsh` and `/usr/share/doc/libvirt-<version-number>` — Contains subcommands and options for the **virsh** virtual machine management utility as well as comprehensive information about the **libvirt** virtualization library API.
- `/usr/share/doc/gnome-applet-vm-<version-number>` — Documentation for the GNOME graphical panel applet that monitors and manages locally-running virtual machines.
- `/usr/share/doc/libvirt-python-<version-number>` — Provides details on the Python bindings for the **libvirt** library. The **libvirt-python** package allows python developers to create programs that interface with the **libvirt** virtualization management library.
- `/usr/share/doc/python-virtinst-<version-number>` — Provides documentation on the **virt-install** command that helps in starting installations of Fedora and Red Hat Enterprise Linux related distributions inside of virtual machines.
- `/usr/share/doc/virt-manager-<version-number>` — Provides documentation on the Virtual Machine Manager, which provides a graphical tool for administering virtual machines.

부록 A. 실습 1

Xen 게스트 설치

목표: RHEL 5(또는 3, 4)와 Windows XP Xen 게스트를 설치합니다.

필요조건: Virtualization 구성요소를 갖춘 Red Hat Enterprise Linux 5.0가 설치된 워크스테이션이 필요합니다.

이 실습에서는 다양한 가상화 도구를 사용하여 RHEL 5(또는 3, 4)와 Win XP Xen 게스트를 설치하고 구성합니다.

실습 과제 1: PAE 지원 확인

사용자 시스템이 PAE를 지원하는지 반드시 확인해야 합니다. Red Hat Virtualization은 반 가상화 게스트를 실행하기 위한 x86_64 또는 ia64 기반 CPU 구조를 지원합니다. i386 게스트를 실행하려면, 시스템에 PAE 확장을 갖춘 CPU가 필요합니다. 대부분 오래된 랩톱(특히 Pentium Mobile 또는 Centrino 기반 랩톱)은 PAE를 지원하지 않습니다.

1. CPU가 PAE를 지원하는지 확인하려면 다음을 입력하시기 바랍니다:

```
grep pae /proc/cpuinfo
```

2. 다음 출력 결과는 CPU가 PAE를 지원하는지 보여줍니다. 명령어 입력 시 아무것도 출력되지 않으면, CPU는 PAE를 지원하지 않습니다. 모든 실습 과제를 수행하려면 PAE 확장을 갖춘 i386 CPU 또는 x86_64 / ia64 시스템이 필요합니다.

```
flags :
  fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat
  clflush dts acpi
  mmx fxsr sse sse2 ss tm pbe nx up est tm2
```

실습 과제 2: **virt-install**을 사용하여 RHEL5 Beta 2 Xen 반 가상 게스트를 설치합니다.

이번 과제에서 **virt-install**을 사용하여 Red Hat Enterprise Linux 5 Beta 2 Xen 게스트를 설치합니다.

1. Red Hat Enterprise Linux 5 Beta 2 Xen 게스트를 설치하려면, 명령 프롬프트에 다음을 입력합니다: **virt-install**
2. 완전 가상화 게스트를 설치하라는 프롬프트가 뜨면, 다음을 입력합니다: **no**
3. 가상 장치 이름으로 **rhel15b2-pv1**을 입력합니다.
4. RAM 할당량으로 500을 입력합니다.
5. 디스크(게스트 이미지)에 **/xen/rhel15b2-pv1.img**를 입력합니다.
6. 디스크(게스트 이미지) 크기에 6을 입력합니다.
7. **yes**를 입력하여 그래픽형식 지원을 활성화합니다.
8. 설치 위치에 **nfs:server:/path/to/rhel15b2**를 입력합니다.

9. 설치가 시작됩니다. 평소와 같이 설치하시기 바랍니다.
10. 설치가 완료되면, `/etc/xen/rhel5b2-pv1`에 다음을 수정하여 편집합니다:
`#vnc=1#vncunused=1sdl=1`
11. 텍스트 편집기를 사용하여 `/etc/inittab`에 다음을 추가하여 편집합니다: **init 5.#id:3:initdefault:id:5:initdefault:**

실습 과제 3: **virt-manager**를 사용하여 RHEL5 Beta 2 Xen 반 가상화 게스트를 설치합니다.

이번 실습에서 **virt-manager**를 사용하여 Red Hat Enterprise Linux 5 Beta 2 Xen 반 가상화 게스트를 설치합니다.

1. Red Hat Enterprise Linux 5 Beta 2 Xen 게스트를 설치하려면, 명령 프롬프트에 다음을 입력합니다: **virt-manager**
2. 오픈 연결 창에서 내부 Xen 호스트를 선택하고 **연결**을 클릭합니다.
3. Start Red Hat's Virtual Machine Manager application, and from the **File** menu, click on **New**.
4. **앞으로**를 클릭합니다.
5. 시스템 이름으로 **rhel5b2-pv2**를 입력하고 **앞으로**를 입력합니다.
6. **반 가상화**를 선택하고 **앞으로**를 클릭합니다.
7. 설치 매체 URL에 **nfs:server:/path/to/rhel5b2**를 입력하고 **앞으로**를 클릭합니다.
8. **Simple File**을 선택하고, 파일 위치에 **/xen/rhel5b2-pv2.img**를 입력합니다. 6000 MB를 선택하고 **앞으로**를 클릭합니다.
9. VM 시작과 최대 메모리로 500을 선택하고 **앞으로**를 클릭합니다.
10. **완료**를 클릭합니다.

가상 장치 콘솔 창이 나타납니다. 평소처럼 설치를 시작하고 완료하시기 바랍니다.

실습 과제 4: Intel-VT 또는 AMD-V 지원 확인

이 실습에서 시스템이 Intel-VT 또는 AMD-V 하드웨어를 지원하는지 반드시 확인해야 합니다. 시스템은 성공적으로 완전 가상화 게스트 운영 체제를 설치하기 위해 Intel-VT 또는 AMD-V가 활성화된 CPU를 지원해야 합니다. Red Hat Virtualization은 일반 HVM 계층과 연동하여 이러한 CPU 제조업체를 지원하고 있습니다.

1. To determine if your CPU has Intel-VT or AMD-V support, type the following command: **egrep -e 'vmx|svm' /proc/cpuinfo**
2. 다음 출력 결과는 CPU가 Intel-VT를 지원하는지 보여줍니다:

```
.flags :
    fpu tsc msr pae mce cx8 apic mtrr mca cmov pat clflush dts acpi
mmx fxsr sse
    sse2 ss ht tm pbe constant_tsc pni monitor vmx est tm2 xtpr
```

명령어 입력 시 아무것도 출력되지 않으면, CPU는 Intel-VT 또는 AMD-V를 지원하지 않습니다.

3. CPU가 Intel-VT 또는 AMD-V를 지원하면, 다음을 입력하시기 바랍니다:

```
cat /sys/hypervisor/properties/capabilities
```

4. The following output shows that Intel-VT support has been enabled in the BIOS. If the command returns nothing, then go into the BIOS Setup Utility and look for a setting related to 'Virtualization', i.e. 'Intel(R) Virtualization Technology' under 'CPU' section on a IBM T60p. Enable and save the setting and do a power off to take effect.

```
xen-3.0-x86_32p hvm-3.0-x86_32 hvm-3.0-x86_32p
```

실습 과제 5: virt-install을 사용하여 RHEL5 Beta 2 Xen 완전 가상화 게스트를 설치합니다.

이 실습에서 **virt-install**를 사용하여 Red Hat Enterprise Linux 5 Beta 2 Xen 완전 가상화 게스트를 설치합니다:

1. Red Hat Enterprise Linux 5 Beta 2 Xen 게스트를 설치하려면, 명령 프롬프트에 다음을 입력합니다: **virt-install**
2. 완전 가상화 게스트를 설치하라는 프롬프트가 뜨면, **yes**를 입력합니다.
3. 가상 장치 이름으로 **rhe15b2-pv2**를 입력합니다.
4. 메모리 할당량에 500을 입력합니다.
5. 디스크(게스트 이미지)로서 **/xen/rhe15b2-fv1.img**를 입력합니다.
6. 디스크(게스트 이미지) 크기에 6을 입력합니다.
7. **yes**를 입력하여 그래픽형식 지원을 활성화합니다.
8. 가상 CD 이미지에 **/dev/cdrom**을 입력합니다.
9. The VNC viewer appears within the installation window. If there is an error message that says "main: Unable to connect to host: Connection refused (111)", then type the following command to proceed: **vncviewer localhost:5900**. VNC port 5900 refers to the first Xen guest that is running on VNC. If it doesn't work, you might need to use 5901, 5902, etc.

설치가 시작됩니다. 평소와 같이 설치하시기 바랍니다.

실습 과제 6: virt-manager을 사용하여 RHEL5 Beta 2 Xen 완전 가상화 게스트를 설치합니다.

이 실습에서 **virt-manager**를 사용하여 Red Hat Enterprise Linux 5 Beta 2 Xen 완전 가상화 게스트를 설치합니다:

1. Red Hat Enterprise Linux 5 Beta 2 Xen 게스트를 설치하려면, 명령 프롬프트에 다음을 입력합니다: **virt-manager**
2. 연결하기 창에서 내부 Xen 호스트를 선택하고 연결을 클릭합니다.
3. Start Red Hat's Virtual Machine Monitor application, and from the **File** menu, click on **New**.
4. 앞으로를 클릭합니다.

5. 시스템 이름으로 **rhel15b2-fv2**를 입력하고 **앞으로**를 클릭합니다.
6. 완전 가상화를 선택하고 **앞으로**를 클릭합니다.
7. CD-ROM 또는 DVD를 지정하고 설치 매체에 대한 경로를 입력합니다. ISO 이미지로부터 설치하려면 ISO 이미지 위치를 지정하고 **앞으로**를 클릭합니다.
8. 간단한 파일을 선택하고, 파일 위치로서 **/xen/rhel15b2-fv2.img**을 입력합니다. 6000 MB를 지정하고 **앞으로**를 클릭합니다.
9. VM 시작과 최대 메모리로 500을 선택하고 **앞으로**를 클릭합니다.
10. **완료**를 클릭합니다.
11. 가상 장치 콘솔 창이 나타납니다.

평소처럼 설치를 완료합니다.

실습 과제 7: **virt-manager**를 사용하여 RHEL3 Xen 완전 가상 게스트를 설치합니다.

이 실습에서 **virt-manager**를 사용하여 Red Hat Enterprise Linux 3 Xen 게스트를 설치합니다:

1. 과제 6의 단계와 동일합니다.

과제 실습 8: **virt-manager**를 사용하여 RHEL4 Xen 완전 가상 게스트를 설치합니다.

이 실습에서 **virt-manager**를 사용하여 Red Hat Enterprise Linux 4 Xen 게스트를 설치합니다:

1. 과제 6의 단계와 동일합니다.

실습 과제 9: **virt-manager**를 사용하여 Windows XP Xen 완전 가상화 게스트를 설치합니다.

이 실습에서 **virt-manager**를 사용하여 Windows XP Xen 완전 가상화 게스트를 설치합니다:

1. Windows XP 호스트에 Red Hat Enterprise Linux 5를 설치하려면 명령 프롬프트에 다음을 입력하시기 바랍니다: **virt-manager**
2. **연결하기** 창에서 내부 Xen 호스트를 선택하고 **연결**을 클릭합니다.
3. Start Red Hat's Virtual Machine Manager application, and from the **File** menu click on **New**.
4. **앞으로**를 클릭합니다.
5. 시스템 이름으로 **winxp**를 입력하고 **앞으로**를 클릭합니다.
6. 완전 가상화를 선택하고 **앞으로**를 클릭합니다.
7. CD-ROM 또는 DVD를 지정하고 설치 매체에 대한 경로를 입력합니다. ISO 이미지로부터 설치하려면 ISO 이미지 위치를 지정하고 **앞으로**를 클릭합니다.
8. 간단한 파일을 선택하고 파일 위치에 **/xen/winxp.img**를 입력합니다. 6000 MB를 지정하고 **앞으로**를 클릭합니다.
9. VM 시작과 최대 메모리로 1024를 지정하고 VCPU에 2를 선택하고 **앞으로**를 클릭합니다.
10. **완료**를 클릭합니다.

11. 가상 장치 콘솔 창이 나타납니다. 평소처럼 설치를 시작하고 완료하시기 바랍니다.
12. **C:** 파티션을 FAT 파일 시스템 형식으로 포맷하도록 선택합니다. Red Hat Enterprise Linux 5는 NTFS 커널 모듈을 지원하지 않습니다. NTFS 파일 시스템 형식으로 포맷한 파티션에는 파일을 Xen 게스트 이미지에 마운트하거나 작성할 수 없습니다.
13. 처음 시스템을 재부팅한 후 **winxp** 게스트 이미지를 편집하시기 바랍니다: **losetup /dev/loop0 /xen/winxp.imgkpartx -av /dev/loop0mount /dev/mapper/loop0p1 /mntcp -prv \$WINDOWS/i386 /mnt/**. 이것은 Windows 설치 시 발생할 수 있는 문제를 해결합니다.
14. 다음을 입력하여 Xen 게스트를 수동으로 재시작합니다: **xm create -c winxp/**
15. 가상 장치 관리자 창에서 **winxp** Xen 게스트를 선택하고 열기를 클릭합니다.
16. 가상 장치 콘솔 창이 나타납니다. 평소처럼 설치를 완료하시기 바랍니다.
17. Whenever a 'Files Needed' dialog box appears, change the path **GLOBALROOT\DEVICE\CDROM0\I386** to **C:\I386**. Depending on your installation, you may or may not see this problem. You may be prompted for missing files during the installation. Changing the path to **C:\I386** should compensate for this problem.
18. Xen 게스트 콘솔이 정지되면 종료 버튼을 클릭하고 다음 사항을 수정합니다:
/etc/xen/winxp:#vnc=1#vncunused=1sdl=1#vcpus=2
19. 단계 14을 반복하고 평소처럼 설치하시기 바랍니다.

부록 B. 실습 2

실시간 이전

목표: 호스트 사이에 실시간 이전을 구성하고 수행합니다.

필요조건: 가상화 플랫폼에서 Red Hat Enterprise Linux 5.0 Beta 2가 설치된 두 대의 워크스테이션이 필요하며, 한 대에는 Fedora Core 6 Xen 게스트가 설치되어야 합니다.

이 실습에서는 두 호스트 사이에서 실시간 이전을 구성하고 수행합니다.

시작 준비

실습 2에서 두 개의 가상화 호스트가 필요합니다: Xen 게스트와 공유 저장공간. 두 가상화 호스트를 UTP 케이블로 연결해야 하며, 하나의 호스트는 NFS를 통하여 공유 저장공간을 익스포트합니다. 성공적인 이전을 위해 두 가상화 호스트를 구성합니다. Xen 게스트는 공유 저장공간에 위치하며, Xen 게스트에서 스트리밍 서버를 설치합니다. 스트리밍 서버가 Xen 게스트에서 아무런 제한없이 작동하는지 확인하여 실시간 이전이 두 가상화 호스트사이에서 실행되도록 합니다. 이 실습에서는 두 가상화 호스트를 **host1**과 **host2**로 부를 것입니다.

과제 1: Xend 구성(두 개의 Xen 호스트 모두)

이번 실습 과정에서 xend를 HTTP와 재배치 서버로 시작하도록 구성합니다. xend 데몬은 디폴트로 HTTP 서버를 시작하지 않으며 **xm**에 대한 UNIX 도메인 소켓 관리 서버를 시작하고 xend와 의사소통을 합니다. 장치 간 실시간 이전을 활성화하려면, 실시간 이전을 지원하도록 설정해야 합니다:

1. **xend-config.sxp** 파일의 백업을 만듭니다:

```
cp -pr /etc/xen/xend-config.sxp /etc/xen/xend-config.sxp.default
```

2. **/etc/xen/xend-config.sxp**에 다음 사항을 수정하여 편집합니다:

```

#(xend-unix-server yes)(xend-relocation-server
  yes)(xend-relocation-port 8002)(xend-relocation-address
  '')(xend-relocation-hosts-allow '')#(xend-relocation-hosts-allow
'^localhost$
  ^localhost\\.localdomain$')
```

3. xend를 재시작합니다: **service xend restart**

과제 2: NFS를 통하여 공유 저장공간 익스포트

이번 실습 과정에서 NFS를 공유 저장공간을 익스포트하도록 구성합니다.

1. **/etc/exports**에 다음 줄을 추가하여 편집합니다: **/xen *(rw, sync, no_root_squash)/**
2. **/etc/exports**를 저장하고 NFS 서버를 재시작하고, NFS가 디폴트로 시작하는지 확인합니다: **service nfs start, chkconfig nfs on.**
3. **host1**에서 NFS 서버를 시작하고 **host2:mount host1:/xen**에 마운트합니다.
4. 이제 **host1**에서 Xen 게스트를 시작하고 **fc6-pv1**(또는 실습1에서 **fc6-pv2**)을 선택합니다:

```
xm create -c fc6-pv1
```

과제 3: Xen 게스트 스트리밍 서버 설치

실습 목적으로 스트리밍 서버(gnump3d)를 설치합니다. gnump3d는 OGG vorbis 파일을 지원하며 설치, 구성 및 수정하기 간편하기에 스트리밍 서버로 사용될 것입니다.

1. <http://www.gnump3d.org/>에서 **gnump3d-2.9.9.9.tar.bz2** 파일을 다운로드합니다. 파일 압축을 풀고 **gnump3d-2.9.9.9/** 디렉토리에서 gnump3d 프로그램을 컴파일하고 설치합니다: **tar xvjf gnump3d-2.9.9.9.tar.bz2cd gnump3d-2.9.9.9/make install**
2. Create a **/home/mp3** directory and copy TruthHappens.ogg from Red Hat's Truth Happens page to **mkdir /home/mp3wget -c http://www.redhat.com/v/ogg/TruthHappens.ogg**
3. 다음 명령어를 입력하여 스트리밍 서버를 시작합니다:

```
command:gnump3d
```

4. 두 Xen 호스트 중 하나에 영화 플레이어를 실행합니다. 영화 플레이어를 실행하기 전에 **totem**과 **iso-codecs rpm**이 없으면 설치하시기 바랍니다. 프로그램 메뉴에서 사운드 & 비디오를 선택하고 영화 플레이어를 클릭합니다.
5. 영화를 클릭하고 위치열기를 선택한 후 <http://guest:8888/TruthHappens.ogg>를 입력합니다.

과제 4: 실시간 이전 수행

1. 두 Xen 호스트에서 **TruthHappens.ogg** 파일을 실행합니다.
2. **host1**로부터 **host2**로 실시간 이전을 수행합니다:

```
xm migrate -live fc6-pv1 host2
```

3. 다음 명령어를 사용하여 두 Xen 호스트에서 다중 터미널 창을 열어보시기 바랍니다:

```
watch -n1 xm list
```

4. 실시간 이전이 시작되는 것을 살펴보시기 바랍니다. 이전이 완료되는 데 얼마나 걸리는지 확인하시기 바랍니다.

도전 과제: Xen 게스트에서 VNC 서버를 구성합니다.

If time permits, from within the Xen guest, configure the VNC server to initiate when **gdm** starts up. Run VNC viewer and connect to the Xen guest. Play with the Xen guest when the live migration occurs. Attempt to pause/resume, and save/restore the Xen guest and observe what happens to the VNC viewer. If you connect to the VNC viewer via **localhost :590x**, and do a live migration, you won't be able to connect to the VNC viewer again when it dies. This is a known bug.

부록 C. REVISION HISTORY

고침 2.0-11.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
고침 2.0-11 Rebuild for Publican 3.0	2012-07-18	Anthony Towns
고침 5.1.0-10 Resolves: #245684 Content Updates	Fri Aug 3 2007	Michael Hideo Smith