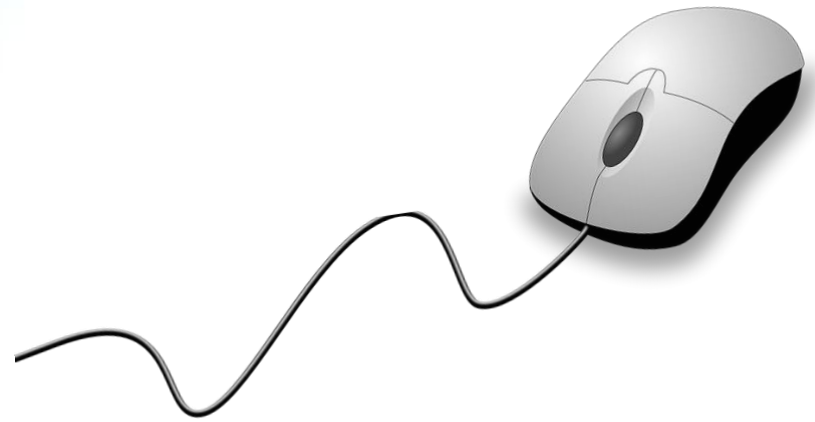


공개SW 솔루션 설치 & 활용 가이드

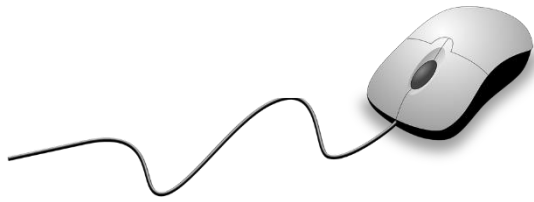
시스템SW > 가상화



제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide



CONTENTS

1. 개요
2. 기능요약
3. 실행환경
4. 설치 및 실행
5. 기능소개
6. 활용예제
7. FAQ
8. 용어정리

1. 개요



소개	<ul style="list-style-type: none"> KVM(Kernel-based Virtual Machine)은 가상화 확장(Intel VT or AMD-V)을 제공하는 x86 하드웨어상의 리눅스에 대한 전가상화(full virtualization) 솔루션 핵심 가상화 인프라를 제공하는 적재 가능한 커널 모듈(kvm.ko)과 프로세서 전용 모듈(intel.ko or kvm-amd.ko)로 구성 및 리눅스 커널 2.6.20 버전부터 메인 커널에 포함 KVM은 Red Hat Enterprise Linux(RHEL), CentOS, Fedora 기반 리눅스 등에 기본적으로 포함되어 있는 hypervisor로 손쉽게 사용 가능 		
주요기능	<ul style="list-style-type: none"> KVM은 커널에서 커널 모듈을 이용하여 hypervisor로 전환한다는 점에서 전가상화 방식 일반적인 리눅스 커널을 사용 리눅스의 하드웨어 확장성, power management, hot-plug 등의 기능 상속 KVM은 특정 장치에 대해 반가상화를 제공하며, SPICE 또는 VMWARE 드라이버를 사용하여 VGA 그래픽 인터페이스 반가상화 지원 		
대분류	시스템SW	소분류	가상화
라이선스형태	GNU GPL 또는 LGPL	사전설치 솔루션	<ul style="list-style-type: none"> Intel : vmx flag AMD : svm fkage
운영제제	Linux	버전	qemu-kvm-1.2.0
특징	<ul style="list-style-type: none"> KVM에서 가상화를 제공하기 위해서는 한 가지 전제 조건이 붙는데, 사용하는 CPU에서 HMM(Hardware Virtual Machine) 기능제공 필요 hypervisor가 물리적인 하드웨어 전체에 직접 접근하지만, 이 hypervisor 자체가 실제 리눅스 커널의 모듈 형태로 제공한다는 점에서 다른 전 가상화에 비해 나은 성능과 가상화 관리 환경 제공 호스트 서버에서 보면 게스트 서버는 하나의 프로세스 간주되므로 overhead가 거의 없어 전가상화와 반가상화의 장점만 모아둔 hypervisor로 평가 Linux, BSD, SOLARIS, WINDOWS, HAIKU, REACTOS, PLAN9, AROS RESERCH OS 등 다양한 Guest OS를 지원하고, QEMU-KVM을 이용하면 MAC OS X 도 지원 		
보안취약점	<ul style="list-style-type: none"> 취약점 ID : CVE-2018-10901 심각도 : 7.8 HIGH(V3) 취약점 설명 : Linux 커널의 KVM 가상화 하위 시스템에 결함이 발견 대응방안 : Red Hat Enterprise Linux 6.6 이상으로 업데이트 참고 경로 : https://access.redhat.com/errata/RHSA-2018:2392 		
개발회사/커뮤니티	Open Virtualization Alliance(OVA)		
공식 홈페이지	www.linux-kvm.org		



2. 기능요약



주요기능	지원여부
KVM은 리눅스 커널을 커널 모듈을 이용하여 hypervisor로 전환한다는 점에서 전가상화 방식이다.	지원
KVM은 윈도우를 게스트로 설치할 수 있으므로 전가상화를 지원한다고 볼 수 있다. 또한 KVM이 설치되지 않은 서버에는 QEMU를 설치하는데 QEMU는 대표적인 반가상화 지원 hypervisor 이며, KVM은 QEMU를 기반으로 돌아가는 hypervisor이므로 반가상화와 전가상화를 모두 지원한다고 볼 수 있다.	지원
KVM은 특정 장치에 대해 반가상화를 제공한다. VMWARE 드라이버를 사용하여 VGA 그래픽 인터페이스 반가상화를 지원한다.	지원
Hypervisor는 물리적인 하드웨어 전체에 직접 접근하지만, 이 hypervisor 자체가 실제 리눅스 커널의 모듈 형태로 제공한다는 점에서 다른 전가상화에 비해 더 좋은 성능과 가상화 관리 환경을 제공한다.	지원
Linux, BSD, SOLARIS, WINDOWS, HAIKU, REACTOS, PLAN9, AROS RESERCH OS 등 다양한 Guest OS를 지원하고, QEMU-KVM을 이용하면 MAC OS X 도 지원한다.	지원

3. 실행환경



1. OS

CentOS Linux release 7.3.1611 (Core) 환경

2. 사전 설치 솔루션

- vmx flag : Intel CPUs
 - svm flag : AMD CPUs
- > 해당 machine의 CPU가 가상화를 지원해야한다.

virt-manager-1.4.0-2.el7.noarch

libvirt-python-2.0.0-2.el7.x86_64

libvirt-client-2.0.0-10.el7_3.9.x86_64

libvirt-2.0.0-10.el7_3.9.x86_64

3. KVM package

qemu-img-1.5.3-126.el7_3.10.x86_64

qemu-kvm-1.5.3-126.el7_3.10.x86_64



4. 설치 및 실행

세부 목차



- 4.1 Verify virtualization support
- 4.2 Installing the KVM modules
- 4.3 Start the service daemon
- 4.4 Verify that the KVM module is loaded
- 4.5 Start the virt-Manager



4. 설치 및 실행



4.1 Verify virtualization support

- CPU가 가상화를 지원하는지 확인

-> vmx flag : Intel CPUs
svm flag : AMD CPUs

-> `grep -E '(vmx|svm)' /proc/cpuinfo`

출력 화면에서 vmx또는 svm이라는 단어를 찾아야 한다.
그렇지 않으면 CPU가 가상화를 지원하지 않는다.

```
[root@localhost ~]# grep -E '(vmx|svm)' /proc/cpuinfo
flags       : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm const
ant_tsc arch_perfmon pebs bts nopl xtopology tsc_reliable nonstop_tsc aperfmperf eagerfpu pni pclmulqdq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe po
pcnt aes xsave avx f16c rdrand hypervisor lahf_lm 3dnowprefetch ida arat epb pln pts dtherm hwp hwp_notify hwp_act_window hwp_epp tpr_shadow vnmi ept vpid fs
gsbase smep xsaveopt xsavec xgetbv1
[root@localhost ~]#
```



4. 설치 및 실행



4.2 Installing the KVM modules(1/3)

- KVM 관련 Package 설치 1

-> yum install qemu-kvm qemu-img

```
[root@localhost ~]# yum install qemu-kvm qemu-img
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: ftp.kaist.ac.kr
 * extras: ftp.kaist.ac.kr
 * updates: ftp.kaist.ac.kr
Resolving Dependencies
--> Running transaction check
---> Package qemu-img.x86_64 10:1.5.3-126.el7 will be updated
---> Package qemu-img.x86_64 10:1.5.3-126.el7_3.10 will be an update
---> Package qemu-kvm.x86_64 10:1.5.3-126.el7 will be updated
---> Package qemu-kvm.x86_64 10:1.5.3-126.el7_3.10 will be an update
--> Processing Dependency: qemu-kvm-common = 10:1.5.3-126.el7_3.10 for package: 10:qemu-kvm-1.5.3-126.el7_3.10.x86_64
--> Running transaction check
---> Package qemu-kvm-common.x86_64 10:1.5.3-126.el7 will be updated
---> Package qemu-kvm-common.x86_64 10:1.5.3-126.el7_3.10 will be an update
--> Finished Dependency Resolution
```



4. 설치 및 실행



4.2 Installing the KVM modules(2/3)

- KVM 관련 Package 설치 2

-> yum install virt-manager libvirt libvirt-python libvirt-client

-> GUI 환경을 설치하지 않았거나 그래픽 환경으로 관리할 생각이 없다면 'virt-manager' 패키지는 제외해도 무관하다.

```
[root@localhost ~]# yum install virt-manager libvirt libvirt-python libvirt-client
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: ftp.kaist.ac.kr
* extras: ftp.kaist.ac.kr
* updates: ftp.kaist.ac.kr
Resolving Dependencies
--> Running transaction check
--> Package libvirt.x86_64 0:2.0.0-10.el7_3.9 will be installed
--> Processing Dependency: libvirt-daemon-driver-storage = 2.0.0-10.el7_3.9 for package: libvirt-2.0.0-10.el7_3.9.x86_64
--> Processing Dependency: libvirt-daemon-driver-secret = 2.0.0-10.el7_3.9 for package: libvirt-2.0.0-10.el7_3.9.x86_64
--> Processing Dependency: libvirt-daemon-driver-qemu = 2.0.0-10.el7_3.9 for package: libvirt-2.0.0-10.el7_3.9.x86_64
--> Processing Dependency: libvirt-daemon-driver-nwfilter = 2.0.0-10.el7_3.9 for package: libvirt-2.0.0-10.el7_3.9.x86_64
--> Processing Dependency: libvirt-daemon-driver-nodedev = 2.0.0-10.el7_3.9 for package: libvirt-2.0.0-10.el7_3.9.x86_64
--> Processing Dependency: libvirt-daemon-driver-network = 2.0.0-10.el7_3.9 for package: libvirt-2.0.0-10.el7_3.9.x86_64
--> Processing Dependency: libvirt-daemon-driver-lxc = 2.0.0-10.el7_3.9 for package: libvirt-2.0.0-10.el7_3.9.x86_64
--> Processing Dependency: libvirt-daemon-driver-interface = 2.0.0-10.el7_3.9 for package: libvirt-2.0.0-10.el7_3.9.x86_64
--> Processing Dependency: libvirt-daemon-config-nwfilter = 2.0.0-10.el7_3.9 for package: libvirt-2.0.0-10.el7_3.9.x86_64
--> Processing Dependency: libvirt-daemon-config-network = 2.0.0-10.el7_3.9 for package: libvirt-2.0.0-10.el7_3.9.x86_64
--> Processing Dependency: libvirt-daemon = 2.0.0-10.el7_3.9 for package: libvirt-2.0.0-10.el7_3.9.x86_64
```



4. 설치 및 실행



4.2 Installing the KVM modules(3/3)

- KVM 관련 Package 설치 3

-> yum groupinstall virtualization-client virtualization-platform virtualization-tools

```
[root@localhost ~]# yum groupinstall virtualization-client virtualization-platform virtualization-tools
Loaded plugins: fastestmirror, langpacks
There is no installed groups file.
Maybe run: yum groups mark convert (see man yum)
Loading mirror speeds from cached hostfile
 * base: ftp.kaist.ac.kr
 * extras: ftp.kaist.ac.kr
 * updates: ftp.kaist.ac.kr
Resolving Dependencies
--> Running transaction check
--> Package libguestfs.x86_64 1:1.32.7-3.el7_3.3 will be installed
--> Processing Dependency: supermin5 >= 5.1.16-2 for package: 1:libguestfs-1.32.7-3.el7_3.3.x86_64
--> Processing Dependency: syslinux-extlinux for package: 1:libguestfs-1.32.7-3.el7_3.3.x86_64
--> Processing Dependency: syslinux for package: 1:libguestfs-1.32.7-3.el7_3.3.x86_64
--> Processing Dependency: scrub for package: 1:libguestfs-1.32.7-3.el7_3.3.x86_64
--> Processing Dependency: hivex for package: 1:libguestfs-1.32.7-3.el7_3.3.x86_64
--> Package virt-install.noarch 0:1.4.0-2.el7 will be installed
--> Package virt-top.x86_64 0:1.0.8-8.el7 will be installed
--> Package virt-viewer.x86_64 0:2.0-12.el7 will be installed
--> Package virt-who.noarch 0:0.17-11.el7_3 will be installed
--> Processing Dependency: python-rhsm >= 1.10.10 for package: virt-who-0.17-11.el7_3.noarch
--> Processing Dependency: python-suds for package: virt-who-0.17-11.el7_3.noarch
--> Processing Dependency: m2crypto for package: virt-who-0.17-11.el7_3.noarch
--> Running transaction check
--> Package hivex.x86_64 0:1.3.10-5.7.el7 will be installed
--> Processing Dependency: perl(Win::Hivex::Regedit) for package: hivex-1.3.10-5.7.el7.x86_64
--> Processing Dependency: perl(Win::Hivex) for package: hivex-1.3.10-5.7.el7.x86_64
```



4. 설치 및 실행



4.3 Start the service daemon

- 가상화 machine 생성을 위한 서비스 데몬(libvirtd)을 실행 및 확인
 - > Systemctl start libvirtd
 - > Systemctl enable libvirtd

```
[root@localhost ~]# systemctl start libvirtd
[root@localhost ~]# systemctl status libvirtd
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2017-08-04 13:34:10 KST; 5min ago
     Docs: man:libvirtd(8)
           http://libvirt.org
  Main PID: 11189 (libvirtd)
    CGroup: /system.slice/libvirtd.service
            └─ 1530 /sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/libexec/libvirt_leaseshelper
               1536 /sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp-script=/usr/libexec/libvirt_leaseshelper
               11189 /usr/sbin/libvirtd

Aug 04 13:34:10 localhost.localdomain systemd[1]: Starting Virtualization daemon...
Aug 04 13:34:10 localhost.localdomain systemd[1]: Started Virtualization daemon.
Aug 04 13:34:10 localhost.localdomain dnsmasq[1530]: read /etc/hosts - 2 addresses
Aug 04 13:34:10 localhost.localdomain dnsmasq[1530]: read /var/lib/libvirt/dnsmasq/default.addnhosts - 0 addresses
Aug 04 13:34:10 localhost.localdomain dnsmasq-dhcp[1530]: read /var/lib/libvirt/dnsmasq/default.hostsfile
[root@localhost ~]# systemctl enable libvirtd
[root@localhost ~]#
```



4. 설치 및 실행



4.4 Verify that the KVM module is loaded

- `lsmod | grep kvm`

```
[root@localhost ~]# lsmod | grep kvm
kvm_intel          170181  0
kvm                554609  1 kvm_intel
irqbypass         13503   1 kvm
[root@localhost ~]#
```

- `yum install -y "@X Window System" xorg-x11-xauth xorg-x11-fonts-* xorg-x11-utils`
-> Minimal 환경으로 os를 설치한 경우에는 virt-manager 가 작동하지 않기 때문에 x-window package를 설치해야 한다.

```
[root@localhost ~]# yum install "@X Window System" xorg-x11-xauth xorg-x11-fonts-* xorg-x11-utils -y
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror.oasis.onnetcorp.com
* extras: ftp.daumkakao.com
* updates: ftp.daumkakao.com
Warning: Group x11 does not have any packages to install.
Package 1:xorg-x11-xauth-1.0.9-1.el7.x86_64 already installed and latest version
Package xorg-x11-fonts-Type1-7.5-9.el7.noarch already installed and latest version
Package xorg-x11-utils-7.5-14.el7.x86_64 already installed and latest version
Resolving Dependencies
```



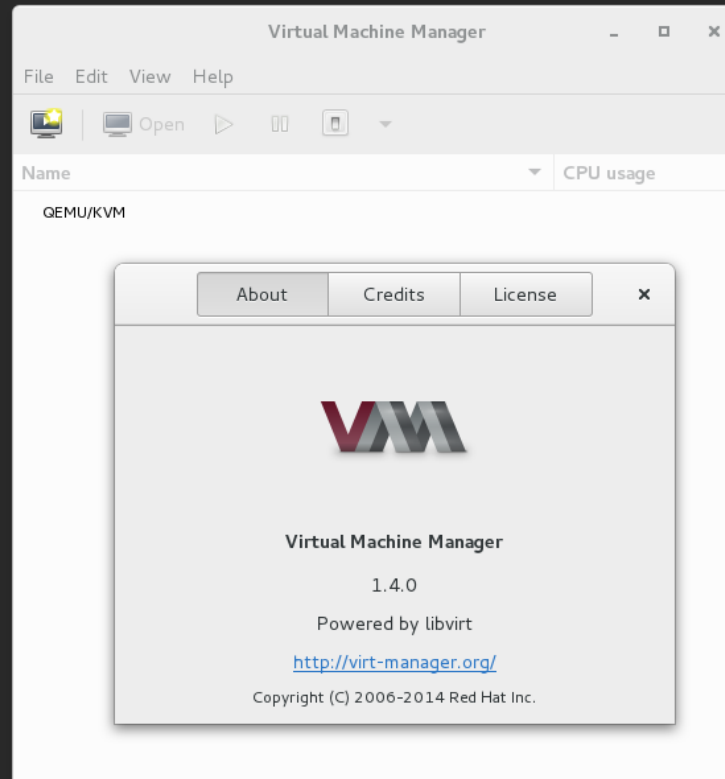
4. 설치 및 실행



4.5 Start the Virt-Manager

- KVM 실행
-> virt-manager

```
[root@localhost ~]# virt-manager  
[root@localhost ~]#
```



5. 기능소개

세부 목차



5.1 Clone

5.1 HardDisk size 변경

5.1 KVM Performance 확인

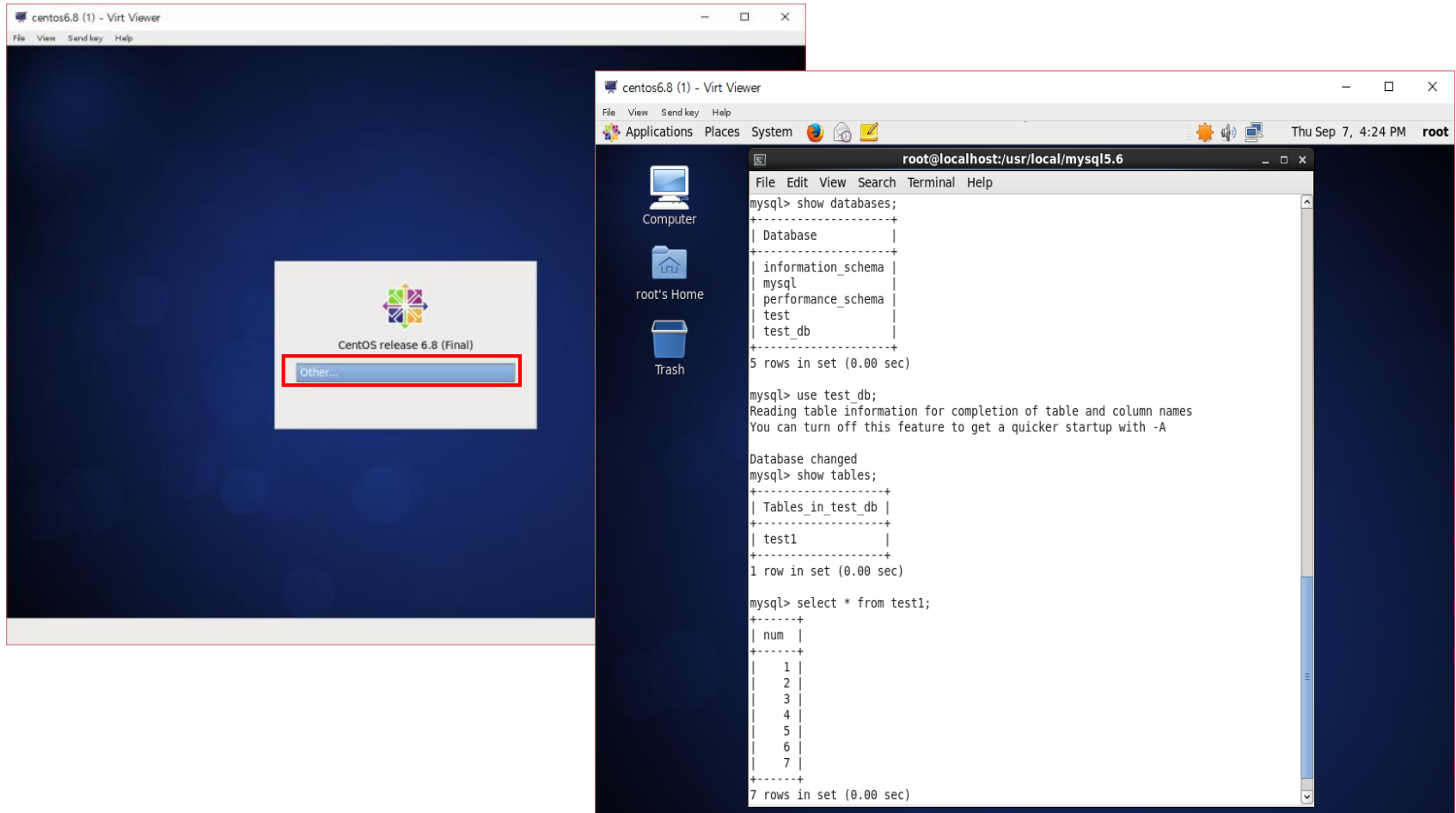


5. 기능소개



5.1 Clone(1/3)

- 구축되어 있는 가상 머신을 복제하여 clone을 생성할 수 있는 기능이다.

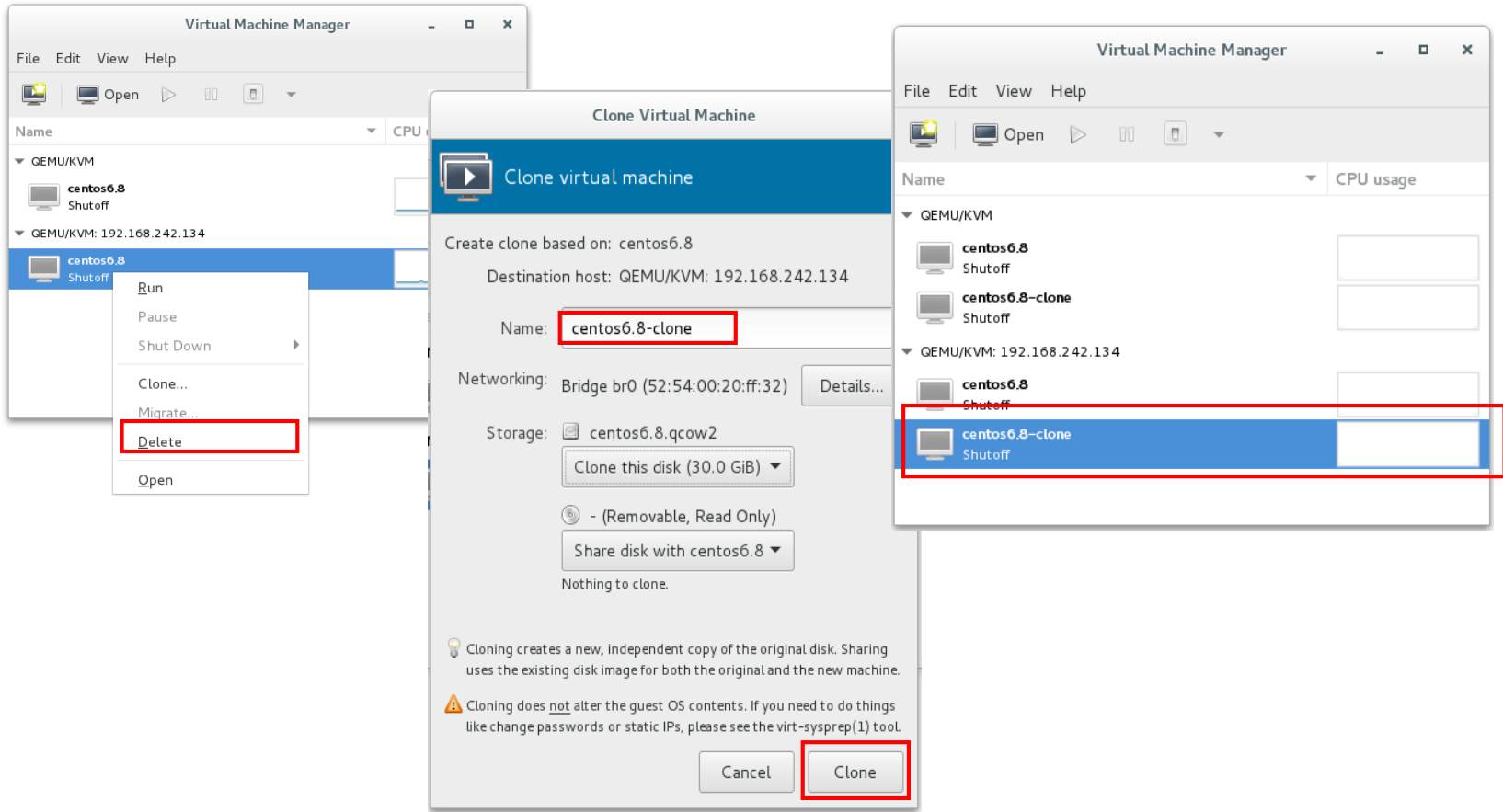


5. 기능소개

5.1 Clone(2/3)



- Virtual Machine Manager에서 Clone 을 선택하여 진행한다.



5. 기능소개

5.1 Clone(3/3)



- Clone 된 가상머신을 확인할 수 있다.

The left screenshot shows the CentOS 6.8 (Final) login screen. The 'mysql' user is selected, highlighted by a red rectangle.

The right screenshot shows a terminal window with the following commands and output:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| test |
| test_db |
+-----+
5 rows in set (0.00 sec)

mysql> use test db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_test_db |
+-----+
| test1 |
+-----+
1 row in set (0.00 sec)

mysql> select * from test1;
+----+
| num |
+----+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |
+----+
7 rows in set (0.00 sec)
```



5. 기능소개



5.2 HardDisk Size 변경

- 해당 directory 는 root 권한이 없으면 열람이 안되기 때문에 root 계정을 써서 접근한다.
 - cd /var/lib/libvirt/images
 - qemu-img resize [크기 조정할 이미지파일 이름] +100G (사용자 임의)

```
[root@localhost ~]# cd /var/lib/libvirt/images/
[root@localhost images]# ls -al
total 14684680
drwx--x--x. 2 root root      4096 Sep  7 16:26 .
drwxr-xr-x. 9 root root      4096 Aug 31 14:13 ..
-rw-----. 1 root root 5289541632 Sep  7 16:59 centos6.8-clone.qcow2
-rw-----. 1 root root 32217432064 Sep  7 16:25 centos6.8.qcow2
-rw-r--r--. 1 qemu qemu 3916431360 Aug 31 15:35 CentOS-6.8-x86_64-bin-DVD1.iso
[root@localhost images]#
```

```
[root@localhost images]# qemu-img resize centos6.8-clone.qcow2 +5GB
Image resized.
```

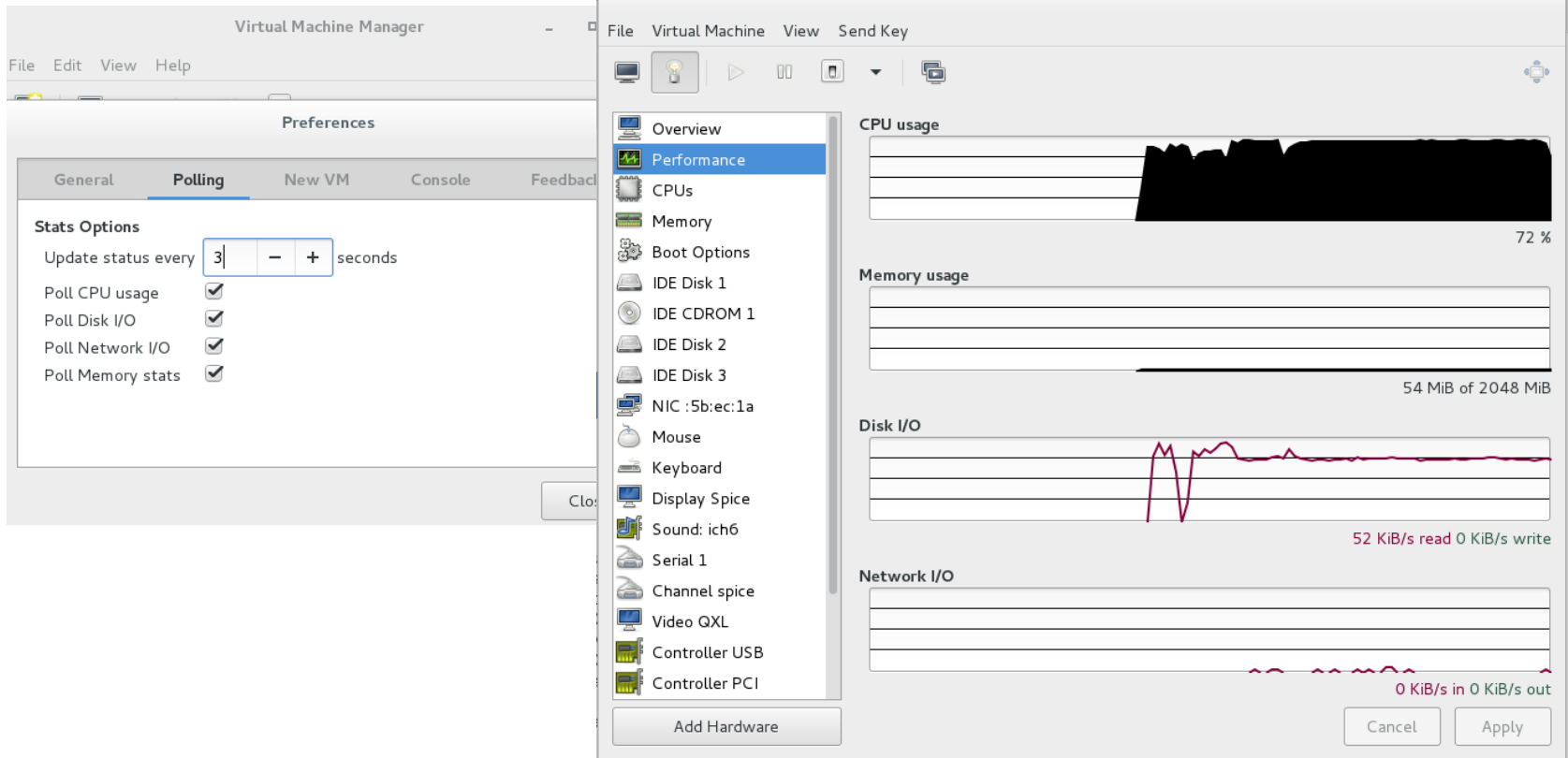
5. 기능소개



5.3 KVM Performance 확인

- 가상 머신의 성능을 확인한다.

- virt-manager -> Edit -> Preferences -> Polling : 사용자가 확인하고 싶은 option 선택 가능하다.



6. 활용예제

세부 목차



6.1 예제 소개

6.2 Configure bridge Interface

6.3 Start Creating Virtual Machines from virt-manager

6.4 Start Creating Virtual Machines from Command Line



6. 활용예제



6.1 예제 소개

- 본 예제는 프로젝트 내부에서 사용할 수 있는 KVM을 사용하여 virtual machine를 구축한다.
- 구축 방법은 총 2가지가 있다. GUI모드인 virt-manage를 사용하여 구축하는 것과 Command Line으로 구축하는 것이다.
- 이 2가지 방법을 모두 사용하여 OS 환경을 구축하는 것을 목표로 한다.

6. 활용예제



6.2 Configure Bridge Interface(1/3)

• Configure Bridge Interface(1/3)

→ hypervisor 네트워크 외부에서 가상 시스템에 접근하려면 Bridge interface 필요하다.

- cd /etc/sysconfig/network-scripts/
- cp ifcfg-ens32 ifcfg-br0 <- ifcfg-ens32 (사용자 별로 다르다.)

```
[root@localhost ~]# cd /etc/sysconfig/network-scripts/  
[root@localhost network-scripts]# cp ifcfg-ens32 ifcfg-br0  
[root@localhost network-scripts]#
```

→ Interface 파일을 편집한다.

- vi ifcfg-ens32
TYPE=Ethernet
BOOTPROTO=static
DEVICE=ens32
ONBOOT=yes
BRIDGE=br0

```
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens32  
TYPE=Ethernet  
BOOTPROTO=static  
DEVICE=ens32  
ONBOOT=yes  
BRIDGE=br0
```



6. 활용예제



6.2 Configure Bridge Interface(2/3)

- **Configure Bridge Interface(2/3)**

→ Bridge 파일 편집한다.

- vi ifcfg-br0
TYPE=Bridge
BOOTPROTO=static
DEVICE=br0
ONBOOT=yes
IPADDR= 사용자IP
NETMASK=255.255.255.0
GATEWAY=사용자Gateway
DNS1=192.168.10.11

```
[root@localhost network-scripts]# cat ifcfg-br0
TYPE=Bridge
BOOTPROTO=static
DEVICE=br0
ONBOOT=yes
IPADDR=192.168.242.134
NETMASK=255.255.255.0
GATEWAY=192.168.242.2
DNS1=192.168.10.11
```

6. 활용예제



6.2 Configure Bridge Interface(3/3)

- **Configure Bridge Interface**

→ bridge interface를 사용하려면 network 서비스를 재 시작한다.

- systemctl restart network

```
[root@localhost network-scripts]# systemctl restart network
```

→ Bridge interface 확인한다.

- ip addr show br0

```
[root@localhost ~]# ip addr show br0  
5: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP qlen 1000  
    link/ether 00:0c:29:c5:7a:51 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.242.134/24 brd 192.168.242.255 scope global br0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::20c:29ff:fec5:7a51/64 scope link  
        valid_lft forever preferred_lft forever
```



6. 활용예제



6.3 Start Creating Virtual Machines from Virt-manager(1/7)

• Start Creating Virtual Machines from Virt-manager(1/7)

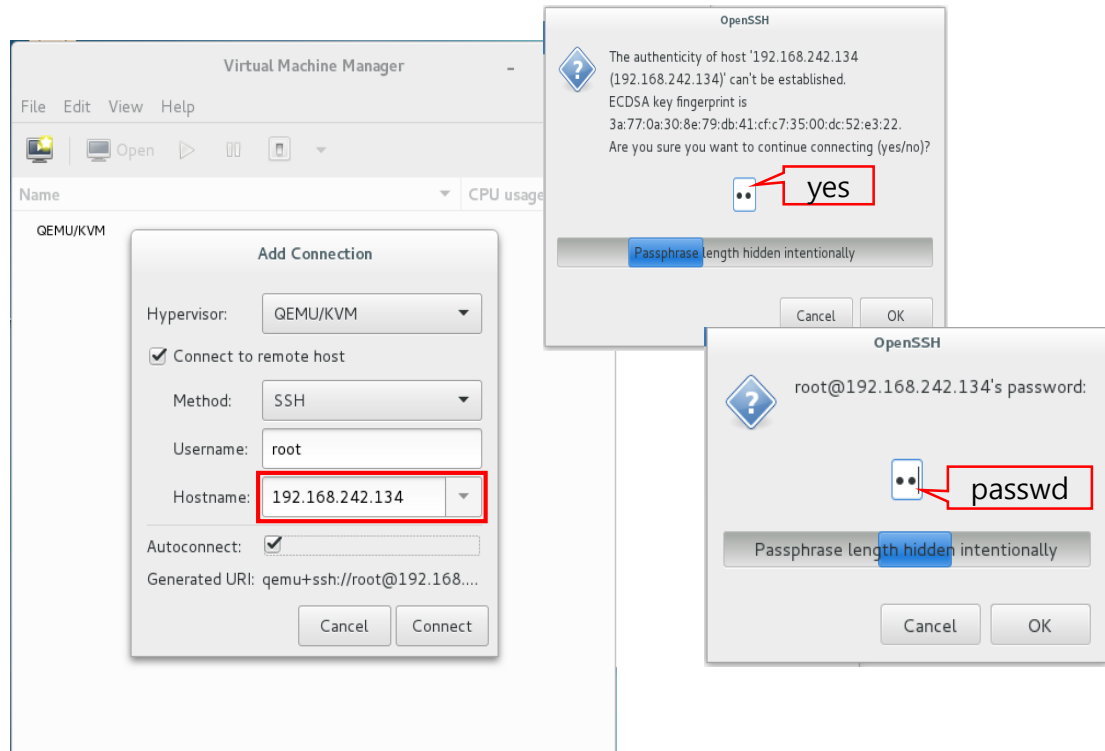
→ 관리 console을 이용하여 원격으로 접속한다.

- terminal창에 virt-manager 입력한다.
- file -> add connection 을 선택하여 원격 설정을 설정한다.

Method : SSH

Username : root

Hostname : host ip



6. 활용예제

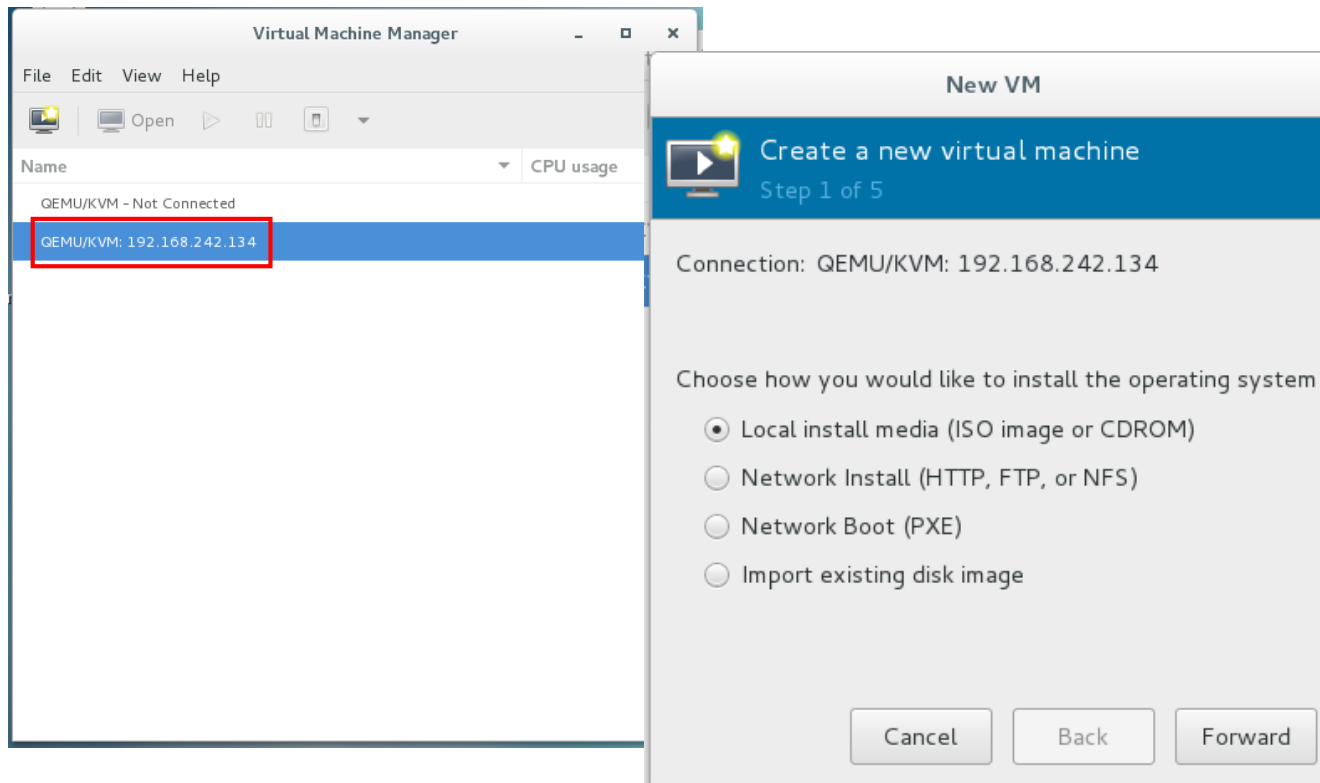


6.3 Start Creating Virtual Machines from Virt-manager(2/7)

- **Start Creating Virtual Machines from Virt-manager(2/7)**

→ New Virtual Machine 선택한다.

→ Local install media 선택한다.



6. 활용예제



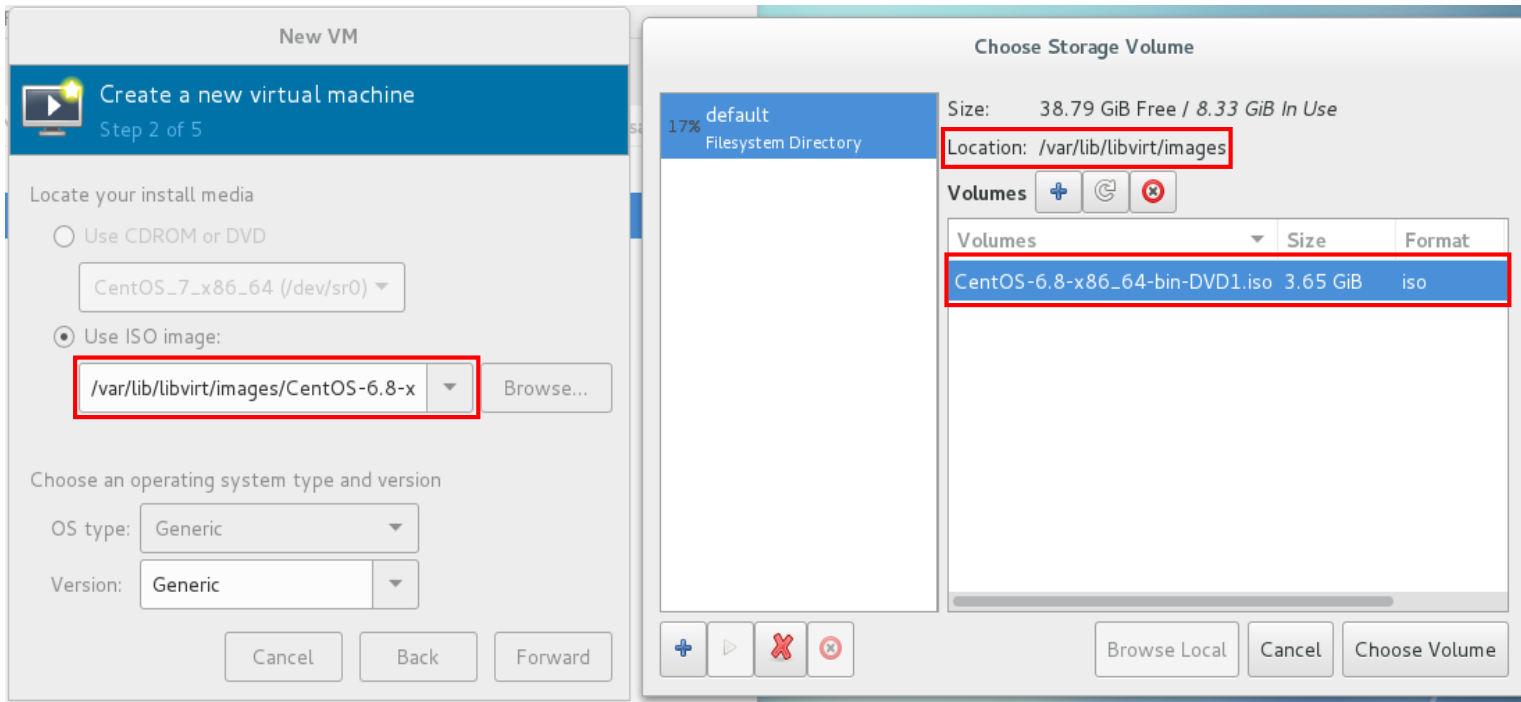
6.3 Start Creating Virtual Machines from Virt-manager(3/7)

• Start Creating Virtual Machines from Virt-manager(3/7)

→ ISO 파일의 경로를 지정한다.

- cd /var/lib/libvirt/images/ 안에 iso파일이 있어야 한다.

```
[root@localhost ~]# cd /var/lib/libvirt/images/  
[root@localhost images]# ll  
total 3824644  
-rw-r--r--. 1 root root 3916431360 Aug 31 15:35 CentOS-6.8-x86_64-bin-DVD1.iso  
[root@localhost images]#
```



6. 활용예제



6.3 Start Creating Virtual Machines from Virt-manager(4/7)

• Start Creating Virtual Machines from Virt-manager(4/7)

- RAM 및 CPU 지정한다.
- Storage size 지정한다.
- 가상 머신의 이름을 지정하고 네트워크를 Bridge br0로 선택한다.

The image displays three overlapping screenshots of the Virt-manager 'New VM' wizard, illustrating the configuration steps for a new virtual machine.

- Step 3 of 5: Choose Memory and CPU settings**
 - Memory (RAM): 2048 MiB (highlighted with a red box)
 - CPUs: 1 (highlighted with a red box)
 - Buttons: Cancel, Back, Forward
- Step 4 of 5: Create a new virtual machine**
 - Enable storage for this virtual machine: ☒
 - Create a disk image for the virtual machine: ☒ (highlighted with a red box)
 - Disk size: 30 GiB (highlighted with a red box)
 - 42.4 GiB available in the default location
 - Select or create custom storage: ☐
 - Buttons: Cancel, Back, Forward
- Step 5 of 5: Ready to begin the installation**
 - Name: centos6.8 (highlighted with a red box)
 - OS: Generic
 - Install: Local CDROM/ISO
 - Memory: 2048 MiB
 - CPUs: 1
 - Storage: 30.0 GiB /var/lib/libvirt/images/centos6.8.qcow2
 - Customize configuration before install: ☐
 - Network selection: Bridge br0: Host device ens32 (highlighted with a red box)
 - Buttons: Cancel, Back, Finish



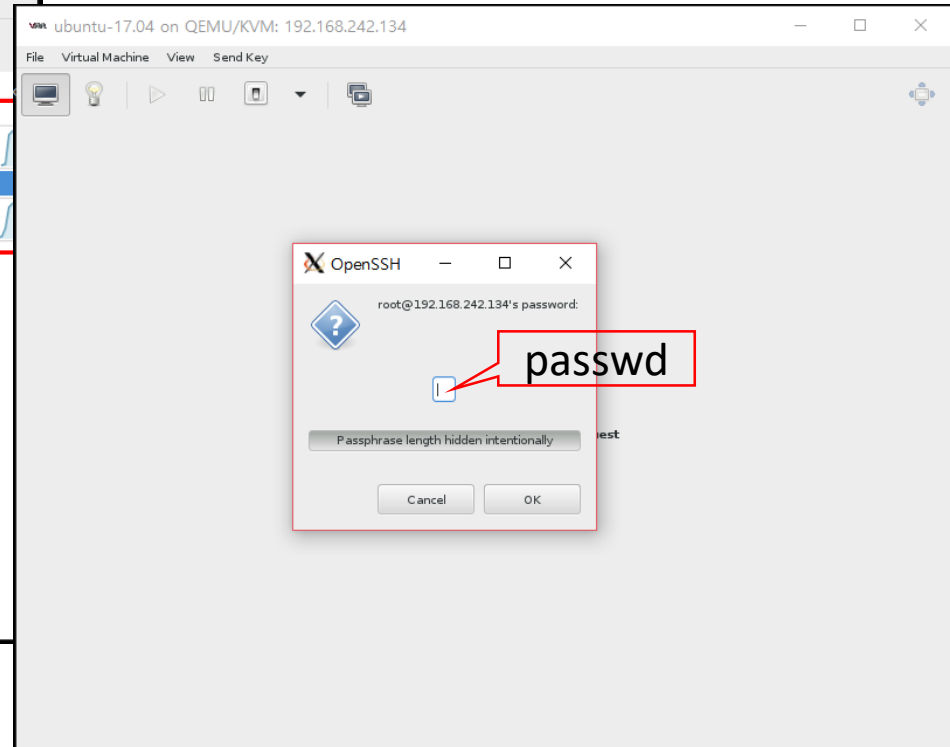
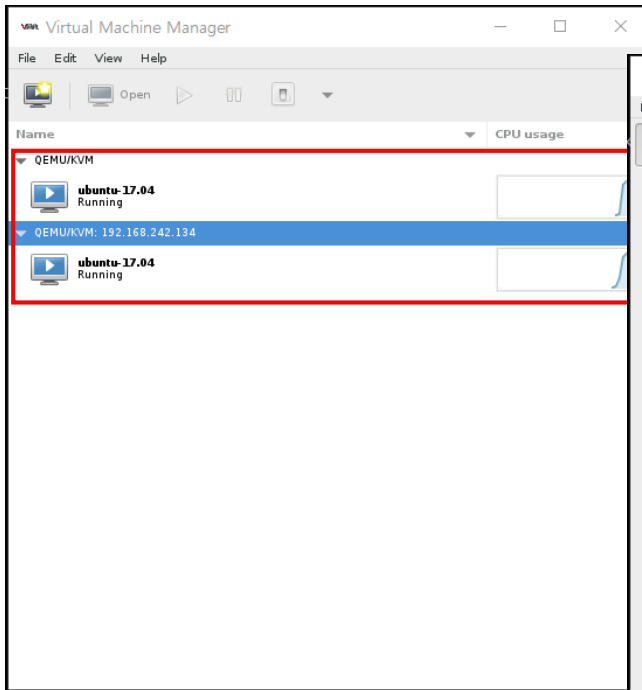
6. 활용예제



6.3 Start Creating Virtual Machines from Virt-manager(5/7)

- Start Creating Virtual Machines from Virt-manager(5/7)

→ 설치를 시작한다.



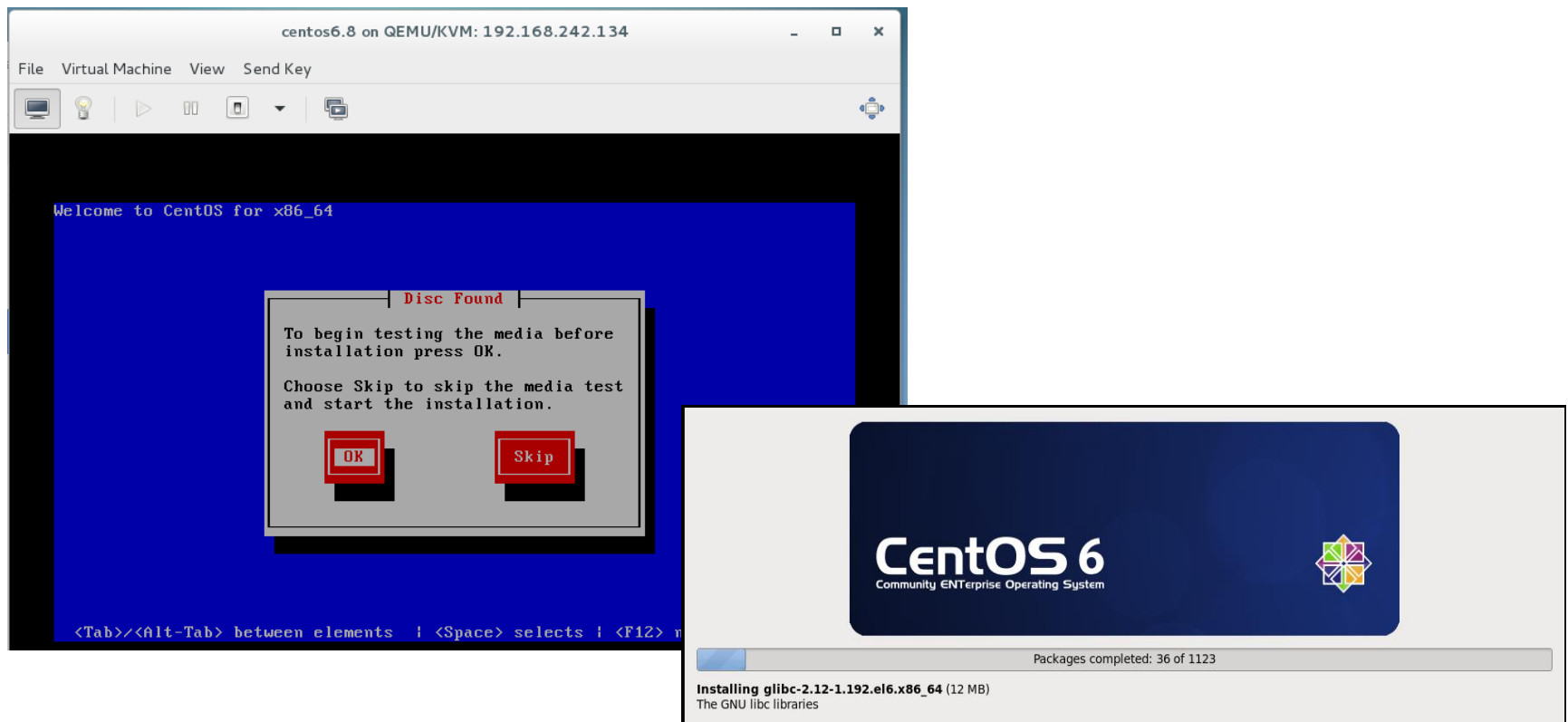
6. 활용예제



6.3 Start Creating Virtual Machines from Virt-manager(6/7)

- Start Creating Virtual Machines from Virt-manager(6/7)

→ 설치 과정



6. 활용예제



6.4 Start Creating Virtual Machines from Command Line(1/3)

- **Start Creating Virtual Machines from Command Line(1/3)**

→ virt-install 명령을 사용하여 console에서 가상 머신을 생성한다.

- virt-install --name=Fedora23 --file=/var/lib/libvirt/images/Fedora23.dsk --file-size=20 --nonsparse --graphics spice --vcpus=2 --ram=2048 --cdrom=/var/lib/libvirt/images/Fedora-Server-DVD-x86_64-23.iso --network bridge=br0 --os-type=linux --os-variant=generic

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# virt-install --name=Fedora23 --file=/var/lib/libvirt/images/  
Fedora23.dsk --file-size=20 --nonsparse --graphics spice --vcpus=2 --ram=2048 --  
cdrom=/var/lib/libvirt/images/Fedora-Server-DVD-x86_64-23.iso --network bridge=b  
r0 --os-type=linux --os-variant=generic  
  
Starting install...  
Allocating 'Fedora23.dsk' | 20 GB 00:00  
Creating domain... | 0 B 00:00  
█
```



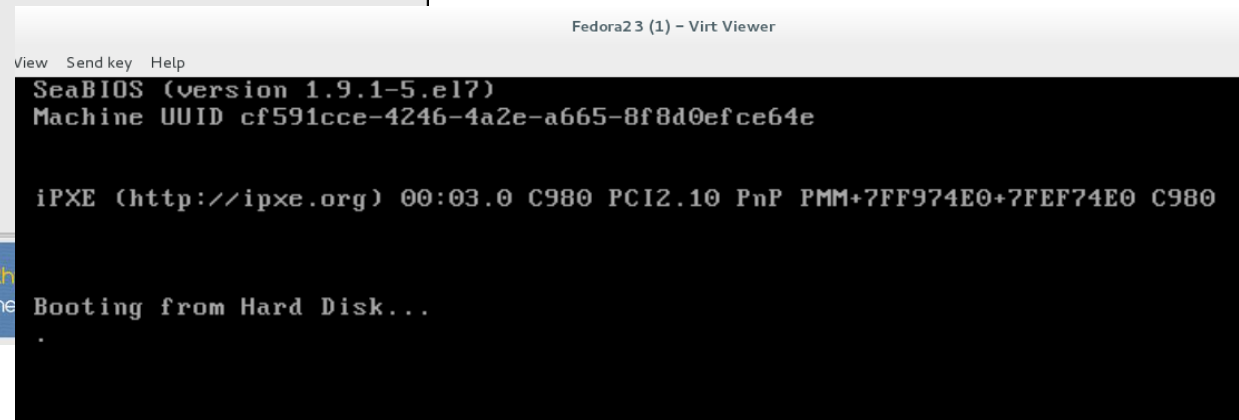
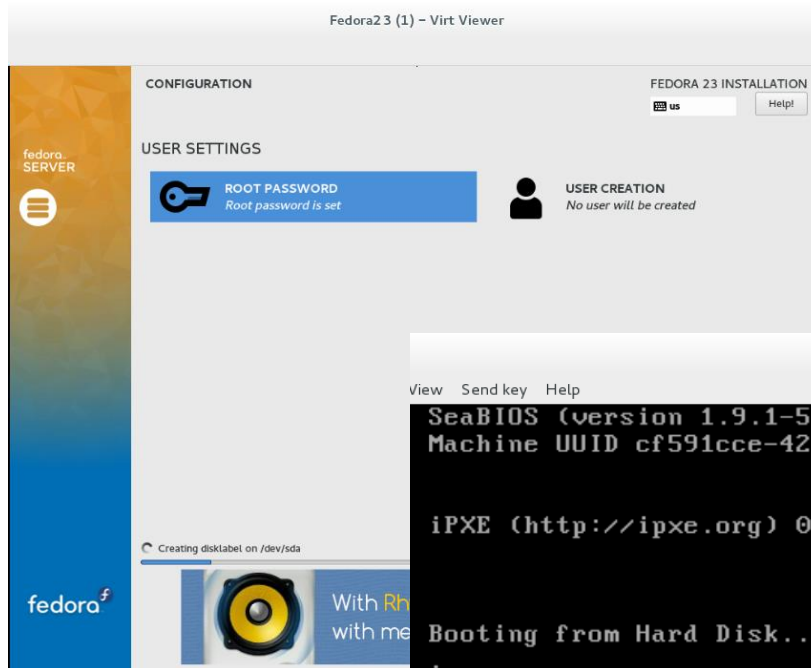
6. 활용예제



6.4 Start Creating Virtual Machines from Command Line(2/3)

- Start Creating Virtual Machines from Command Line(2/3)

→ 설치 시작



6. 활용예제



6.4 Start Creating Virtual Machines from Command Line(3/3)

- Start Creating Virtual Machines from Command Line(3/3)

→ 설치 완료

```
Fedora23 (1) - Virt Viewer

key Help

Fedora 23 (Server Edition)
Kernel 4.2.3-300.fc23.x86_64 on an x86_64 (tty1)

Admin Console: https://192.168.242.140:9090/ or https://[fe80::5054:ff:fe77:36bb]:9090/

localhost login: root
Password:
[root@localhost ~]# ifconfig
ens3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.242.140 netmask 255.255.255.0 broadcast 192.168.242.255
    inet6 fe80::5054:ff:fe77:36bb prefixlen 64 scopeid 0x20<link>
    ether 52:54:00:77:36:bb txqueuelen 1000 (Ethernet)
    RX packets 65 bytes 5329 (5.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 33 bytes 3232 (3.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 0 (Local Loopback)
    RX packets 4 bytes 340 (340.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 340 (340.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]# cat /etc/redhat-release
Fedora release 23 (Twenty Three)
[root@localhost ~]# _
```





Q KVM과 Xen의 차이점은 무엇입니까?

&

A Xen 은 외부 hypervisor로, machine의 제어를 담당하며 guest 사이의 자원들을 분할하는 역할을 수행합니다. 반면, KVM은 Linux의 일부이며 일반적인 Linux 스케줄러 및 메모리 관리 방법을 사용합니다. 이는 KVM이 사용하기 쉽고, KVM만의 특색이 있다는 것을 의미합니다.(ex. KVM은 RAM을 비우기 위해 disk를 guest로 스왑할 수 있습니다.)

또한 KVM은 x86 HVM(vt/svm 명령어 세트)를 지원하는 프로세서에서만 실행되며, Xen은 반 가상화라고 불리는 기술을 사용하여 non-HVM x86 프로세서에서 수정된 운영체제를 실행할 수 있습니다. KVM은 CPU의 반 가상화를 지원하지 않지만 I/O 성능을 향상시키기 위해 장치 드라이버의 반 가상화를 지원할 수 있습니다.

7. FAQ



Q KVM 과 VMware의 차이점은 무엇입니까?

&

A VMware는 유료 제품이지만, KVM은 GPL에 속한 무료 소프트웨어입니다.

Q KVM은 어떤 커널 버전에서 동작합니까?

&

A 사용중인 KVM 버전에 따라 다르며, KVM의 최신 버전은 최신 커널(2.6.17 이상)에서 동작하며 구버전은 이전 커널버전 에서도 동작한다.

8. 용어정리



용어	설 명
전가상화	<ul style="list-style-type: none">• 전가상화는 hypervisor를 이용해 호스트 서버 장치가 직접 가상화되어 가상 머신이 직접 장치를 사용할 수 있도록 하는 방식이다.• 하드웨어를 모두 가상화하여 Hosting되는 운영체제의 수정 없이 다양한 OS 지원 가능하다.• 전가상화가 가능하려면 우선 가상 머신이 호스트 서버의 CPU에 직접 접근이 가능하여 장치의 사용을 명령해야 한다. 따라서, CPU가 전가상화를 지원하는 VT(Virtualization Technology)를 지원해야 한다.
반가상화	<ul style="list-style-type: none">• 소프트웨어 인터페이스를 근본 하드웨어와 완전히 똑같지는 않지만 근본 하드웨어와 비슷한 가상 머신에 제공하는 가상화 기술이다.
QEMU	<ul style="list-style-type: none">• 가상화 소프트웨어 가운데 하나이며, x86 이외의 기종을 위해 만들어진 소프트웨어 스택 전체를 가상머신 위에서 실행할 수 있다
하이퍼바이저(hypervisor)	<ul style="list-style-type: none">• 가상화 기술에서, hypervisor는 단일 시스템에서 여러 운영체제(또는 동일한 운영체제의 여러 인스턴스)를 관리하는 소프트웨어 프로그램이다.• Hypervisor는 시스템의 프로세서, 메모리 및 기타 자원들을 관리하여 각 운영체제에 필요한 부분을 할당한다.• Hypervisors는 특정 프로세서 아키텍처를 위해 설계되었으며, 가상화 관리 자라고도 한다.

Open Source Software Installation & Application Guide

nipa 공개SW역량프라자



이 저작물은 크리에이티브 커먼즈 [저작자표시-비영리-동일조건 변경허락 2.0 대한민국 라이선스]에 따라 이용하실 수 있습니다.