

KVM 기반

# 온프레미스 서비스 구축

추지나

# 목차

## I 프로젝트 소개

1. 프로젝트명
2. 개요
3. 목적
4. 구축방법

## II 프로젝트 구축(상세설명)

1. 물리적인 네트워크 망 설계 및 구축
2. KVM 구축
3. KVM Virt-Manager 구현
4. Libvirt를 이용한 가상 네트워크 구축
5. VM 콘솔을 이용한 원격 접속
6. NFS를 이용한 VM 마이그레이션 실습
7. HTML5기반 Kvm/Libvirt 웹관리도구 “Kimchi” 구축

## III 느낀점

# 1. 프로젝트소개

## 프로젝트명

- 온프레미스 Private Cloud 서비스 구축 (KVM 기반)

## 개요

- 리눅스 기반의 가상화 기술인 KVM을 이용하여 소규모의 프라이빗 클라우드 (Private Cloud) 서비스 망을 구축하고 운영할 수 있도록 한다.
- 연습용 프라이빗 클라우드 서비스 망을 설계하고 이를 바탕으로 KVM을 사용하여 여러 VM을 구축한다.

## 목적

- KVM 기반의 가상네트워크 서비스 망 설계
- KVM 기반의 가상네트워크 서비스 망 구축

## 구축방법

- 1단계 : 물리적 네트워크 망을 설계 및 구축
- 2단계 : KVM 구축 (멀티노드)
- 3단계 : KVM 기반의 Virt-Manager 구축
- 4단계 : NFS 이용한 VM 마이그레이션 설정

# 1. 물리적 네트워크 망 구축

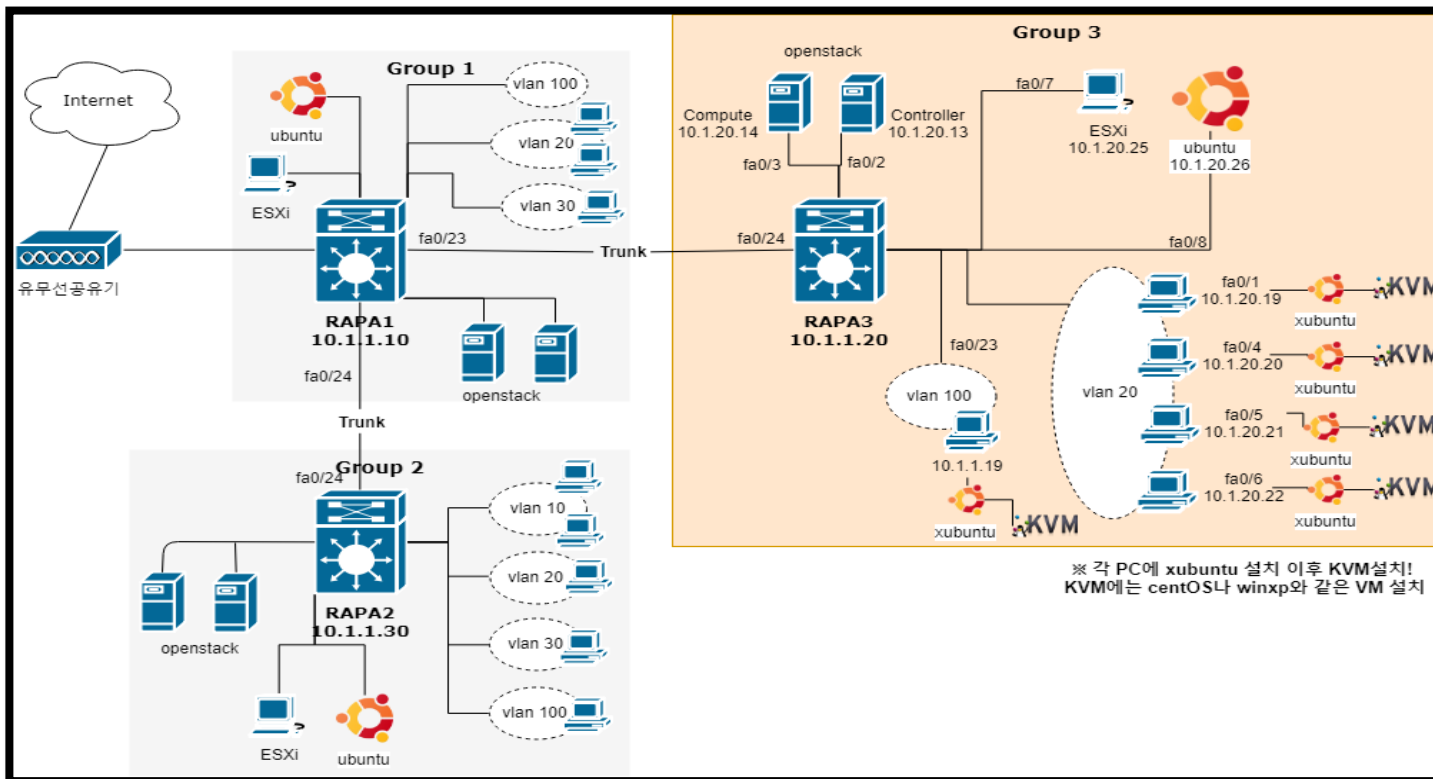
# 상세설명

## 1. 물리적인 네트워크 망 설계 및 구축

### 1) 물리적인 네트워크 망 설계

- 네트워크 구조 설계 (백본 스위치 기반)

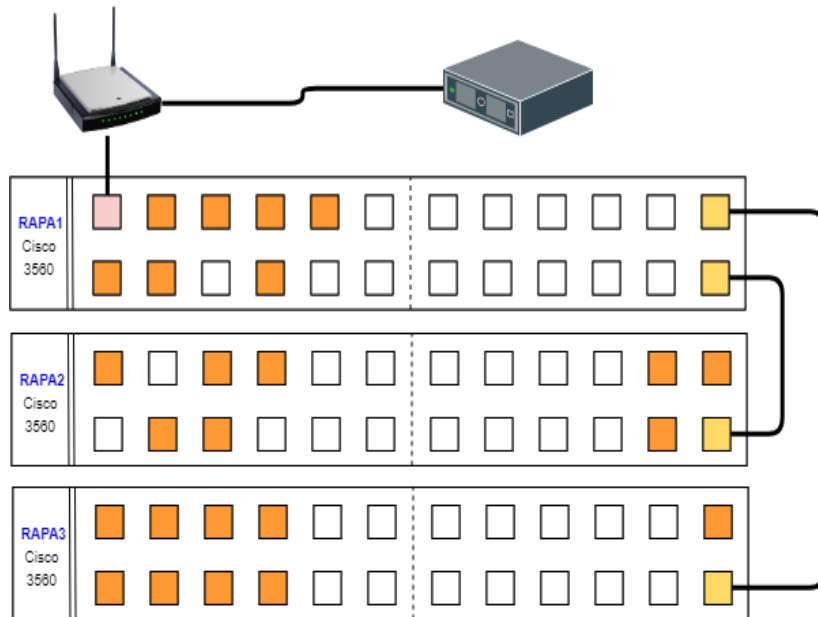
- Catalyst 3560 스위치를 백본(코어 스위치)으로 사용하여 네트워크를 구성함
- Catalyst 3560 스위치에 추가로 Catalyst 3560 스위치를 연결하여 서버 팜을 구성함
- 백본 스위치에 일반 사용자를 위한 Catalyst 3560 스위치를 연결하여 Access망을 구성함.



- 네트워크 논리 설계 (IP Address 및 VLAN 구성)

- 일반 사용자를 위한 IP Address와 VLAN 부여함 → IP Address는 10.1.10.0/28, 10.1.10.16/28, 10.1.10.32/28을 사용  
(+ VLAN은 VLAN 10, VLAN 40, VLAN 50을 사용)
- 외부망 연동을 위한 IP Address와 VLAN 부여함 → IP Address는 192.168.7.150, VLAN은 VLAN 1000 사용
- 디폴트 게이트웨이는 IP Address 192.168.7.77을 사용함.

. 백본 스위치는 RAPA1으로 RAPA2/RAPA3 스위치를 trunk로 연동하였다.



장비명	VLAN 주소(번호)	IP 주소
RAPA1	10.1.20.1(20)	10.1.20.10
	10.1.20.1(20)	10.1.20.20
	10.1.1.10(100)	10.1.1.100
RAPA2	10.1.10.1(10)	10.1.10.13
	10.1.20.1(20)	10.1.20.66
	10.1.30.1(30)	10.1.30.56
RAPA3	10.1.1.10(100)	10.1.1.41
	10.1.20.1(20)	10.1.20.19
	10.1.20.1(20)	10.1.20.20
	10.1.20.1(20)	10.1.20.21
	10.1.20.1(20)	10.1.20.22
	10.1.1.10(100)	10.1.1.19

# 스위치 설정

RAPA1 스위치



RAPA2 스위치



RAPA3 스위치



TRUNK

## 1. RAPA1 스위치 설정

스위치 명	포트 번호	vlan	trunk	subnet
RAPA1	fast ethernet 0/1-3	10		10.1.10.2/28
	fast ethernet 0/4-6	20		10.1.20.1/24
	fast ethernet 0/7-9	30		10.1.30.1/24
	fast ethernet 0/10-12	40		10.1.10.17/28
	fast ethernet 0/13-15	50		10.1.10.33/28
	fast ethernet 0/16-18	60		10.1.10.49/28
	fast ethernet 0/19	1000		192.168.7.150
	fast ethernet 0/20-22	100		10.1.1.10/24
	fast ethernet 0/23-24		O	

## 2. RAPA2 스위치 설정

스위치 명	포트 번호	vlan	trunk	subnet
RAPA2	fast ethernet 0/1-3	40		10.1.10.18/24
	fast ethernet 0/4-6, 0/21-22	20		10.1.20.1/24
	fast ethernet 0/7-9	30		10.1.30.1/24
	fast ethernet 0/23	100		10.1.1.10/24
	fast ethernet 0/24		O	

## 3. RAPA3 스위치 설정

스위치 명	포트 번호	vlan	trunk	subnet
RAPA3	fast ethernet 0/1-10	20		10.1.20.1/24
	fast ethernet 0/11-14	50		10.1.10.34/24
	fast ethernet 0/23	100		10.1.1.10/24
	fast ethernet 0/24		O	

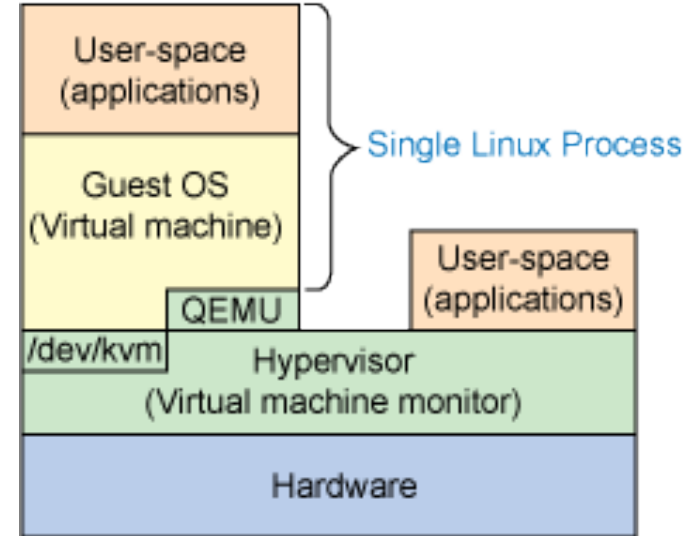


## 2. KVM 구축

# KVM 서비스 구현

## KVM 이란?

- KVM(Kernel-based Virtual Machine: 커널 기반 가상 머신)은 Linux에 구축되는 오픈소스 가상화 기술이다. 구체적으로 설명하자면 KVM을 통해 Linux를 타입1(베어메탈) 하이퍼바이저로 전환하여 호스트 머신이 게스트 또는 VM(가상 머신) 등 독립된 가상 환경 여러 개를 실행할 수 있다.
- 하드웨어 가상화 지원기능 (VT-x, AMD-V)를 활용한 최신 하이퍼 바이저로 KVM 커널 모듈 설치로 하이퍼바이저로 변환이 가능하다.
- 리눅스 표준 커널에 KVM 커널 모듈을 추가하여 사용하며 표준 커널의 메모리 지원, 스케줄러 등을 사용가능한 장점이 있다. 리눅스 컴포넌트 최적화는 하이퍼바이저와 리눅스 Guest OS 둘 다 이점을 갖는다.



## 2. KVM 구축

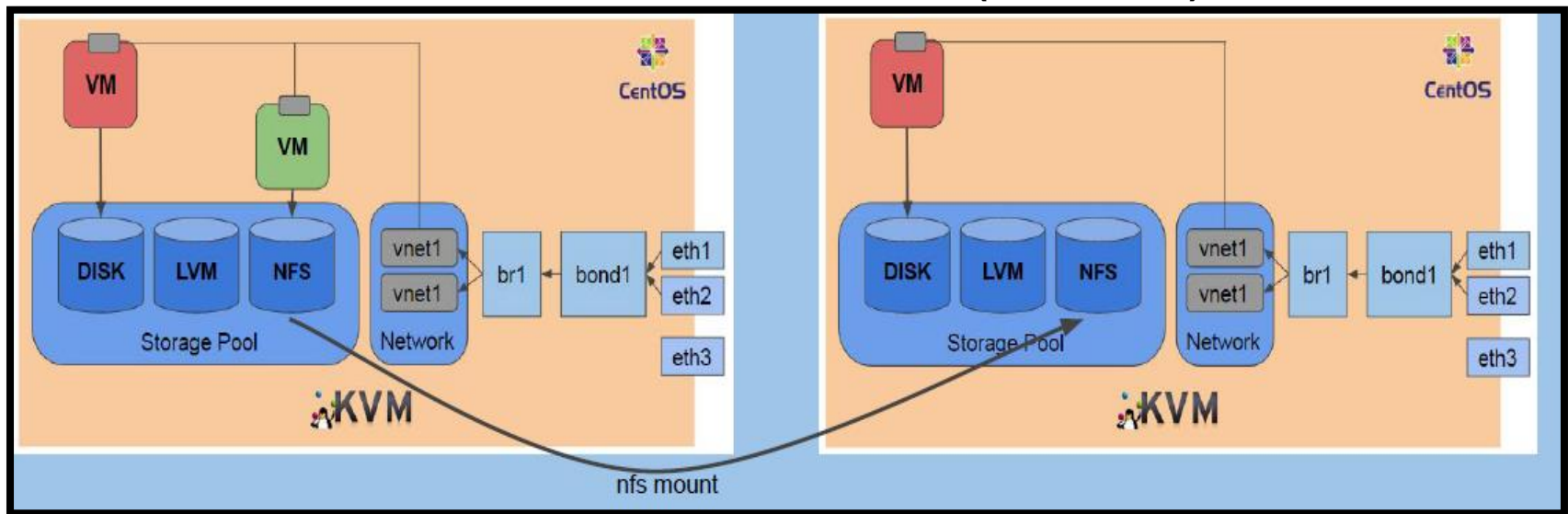
### 1) KVM 구성

- KVM 가상서버 구성도 (Multi Node 구성)

- PC의 Vmware에 KVM 하이퍼바이저 CentOS7 설치
- VM 마이그레이션 Live 마이그레이션 실습을 위한 NFS이용 공유 스토리지 구성

HOST1(10.1.10.36)

HOST2(10.1.10.47)



## KVM 하드웨어 구성

### - KVM 가상 서버 사양

- CPU : i5 (Dual Core, INTEL-VT 지원)
- Memory : 8Gbyte
- SSD : 1Tbyte
- NIC : 1Gbps
- OS : CentOS 7.3 minimal

## KVM 소프트웨어 구성

- KVM 기반 OS : Ubuntu 16.04 server
- 하이퍼바이저 관리도구 : Virtualization Host, KVM\_libvirt, virt\_manager, virt\_viewer 등등...

명령	Guest 관리	Guest 모니터링	Host, 하이퍼바이저	가상 네트워크	스토리지
리스트	Start, stop	Memstat, cpustat	Capabilities, nodeinfo	Net-list, net-define	Pool-list, pool-define

```
[root@localhost ~]# lscpu
Architecture:      x86_64
CPU op-mode(s):    32-bit, 64-bit
Byte Order:        Little Endian
CPU(s):            4
On-line CPU(s) list: 0-3
Thread(s) per core: 1
Core(s) per socket: 1
Socket(s):         4
NUMA node(s):     1
Vendor ID:         GenuineIntel
CPU family:        6
Model:            78
Model name:        Intel(R) Core(TM) i5-6200U
CPU @ 2.30GHz
Stepping:         3
CPU MHz:          2400.139
CPU max MHz:       0.0000
CPU min MHz:       0.0000
BogoMIPS:         4800.27
Virtualization:    VT-x
Hypervisor vendor: VMware
Virtualization type: full
L1d cache:        32K
L1i cache:        32K
L2 cache:         256K
L3 cache:         3072K
```

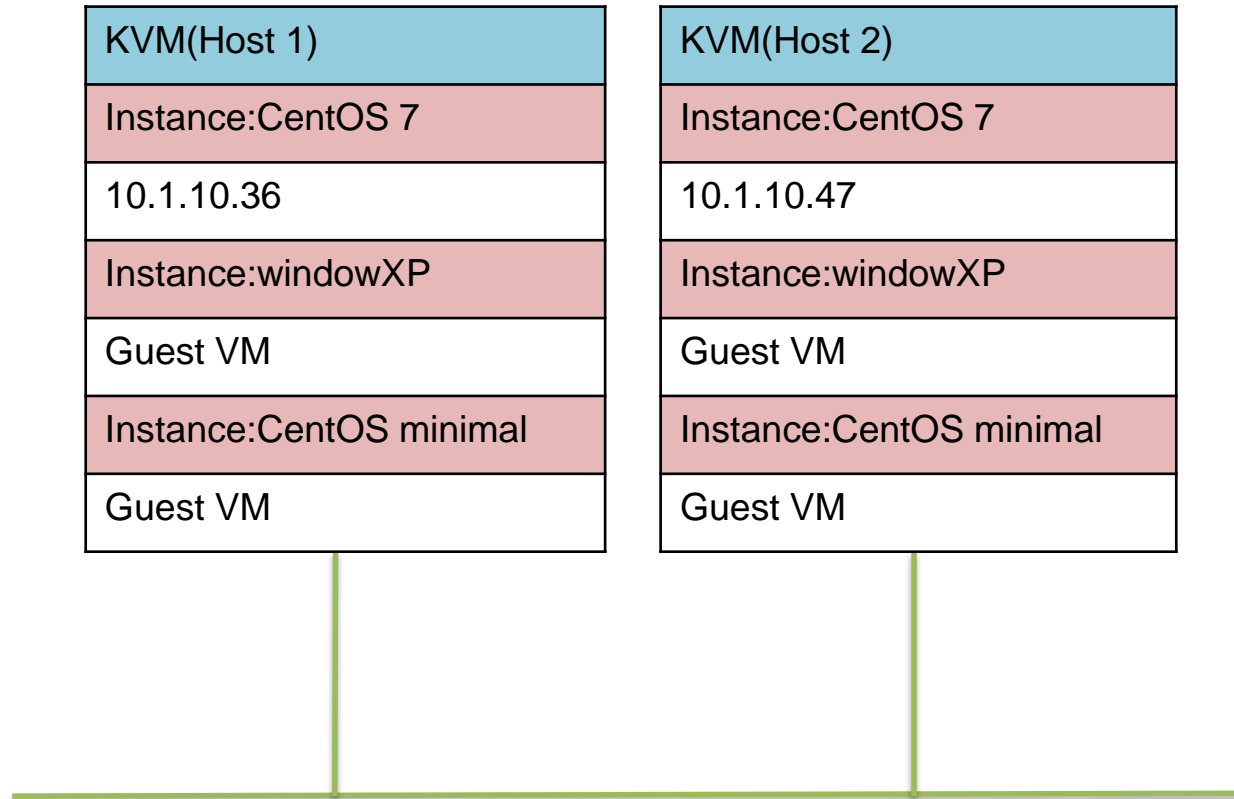
## 3. KVM Virt-Manager 구현

# KVM Virt-Manager(가상화 관리도구)

## 1) KVM 인스턴스 및 서비스 구성

- kvm 인스턴스 및 서비스를 먼저 구성하기 위해 구성도를 만든다.

➢ 총 2대의 호스트를 통해 먼저 각 호스트마다 KVM을 각각 설치한 뒤, CentOS 7 기반의 인스턴스를 각각 만들어주었다.

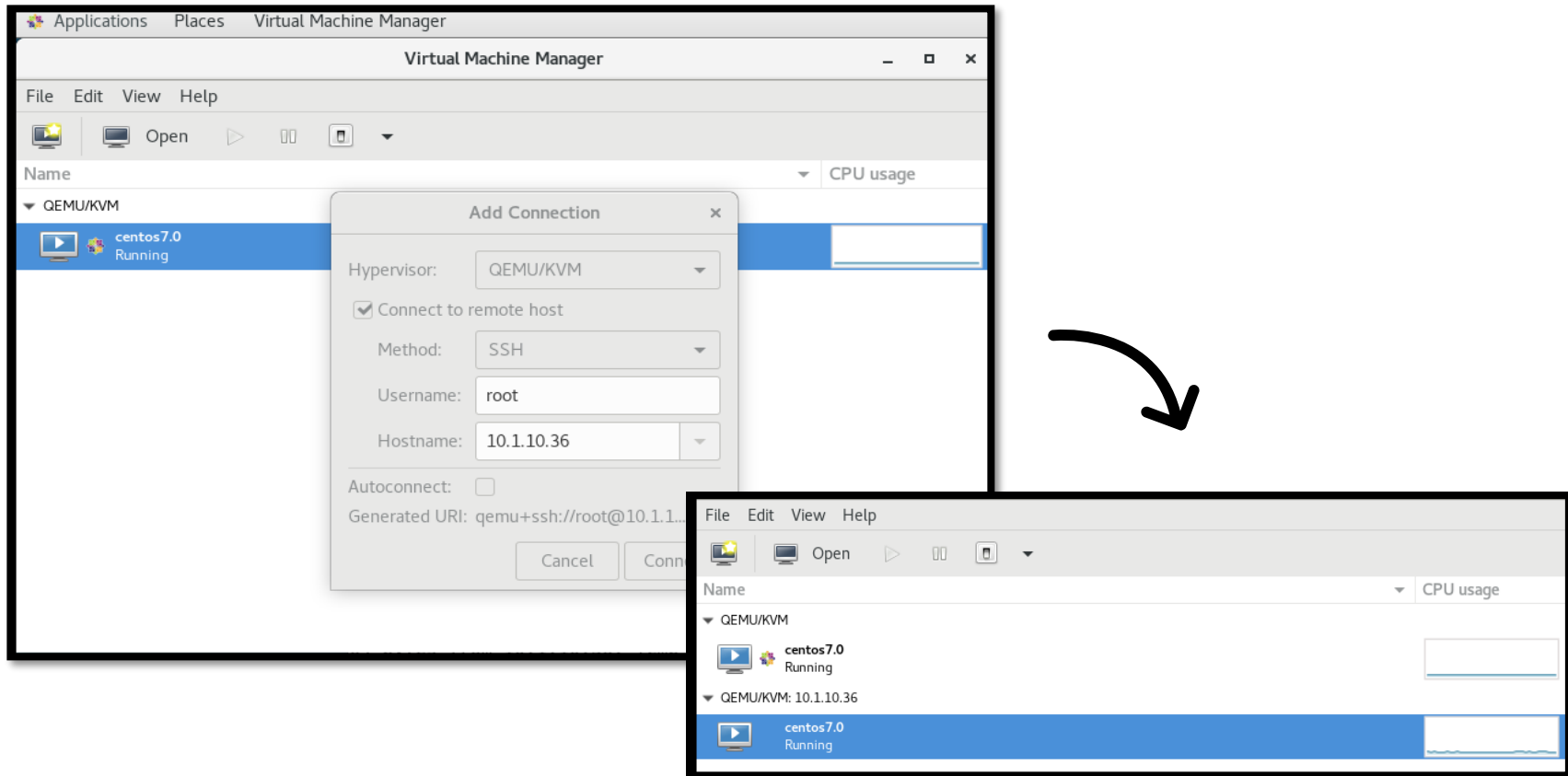


# KVM Virt-Manager(가상화 관리도구)

## 1) KVM 원격 Hypervisor 연결

- 10.1.10.36 를 원격 host로 연결한다.

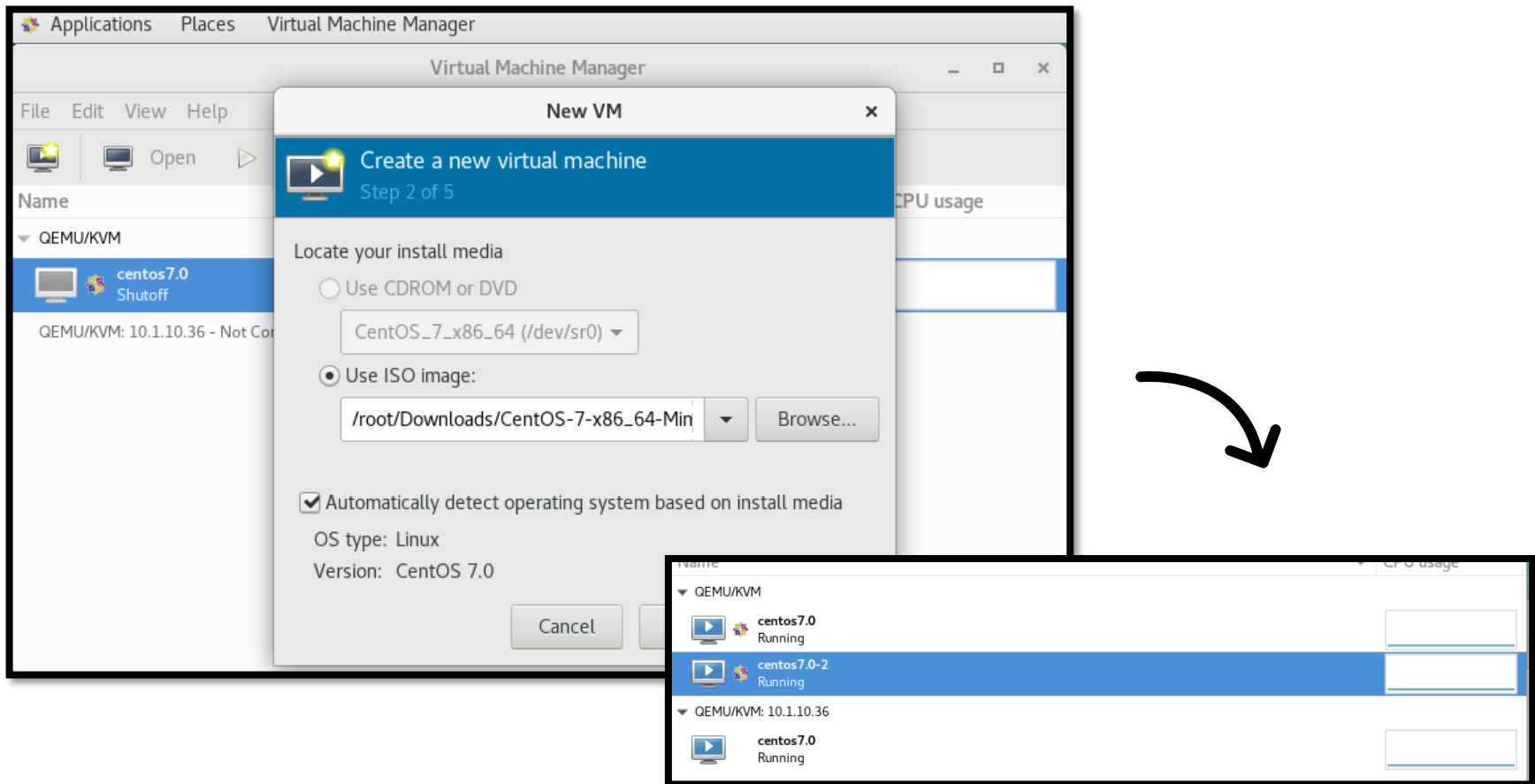
➢ 총 2대의 호스트를 통해 먼저 각 호스트마다 KVM을 각각 설치한 뒤, CentOS 7 기반의 인스턴스를 각각 만들어주었다.



# KVM Virt-Manager(가상화 관리도구)

## 2) KVM virt manager

- virt manager를 이용하여 VM(Cent OS Minimanl) 생성
  - VM을 생성하고 해당 VM이 생겼는지 확인한다.

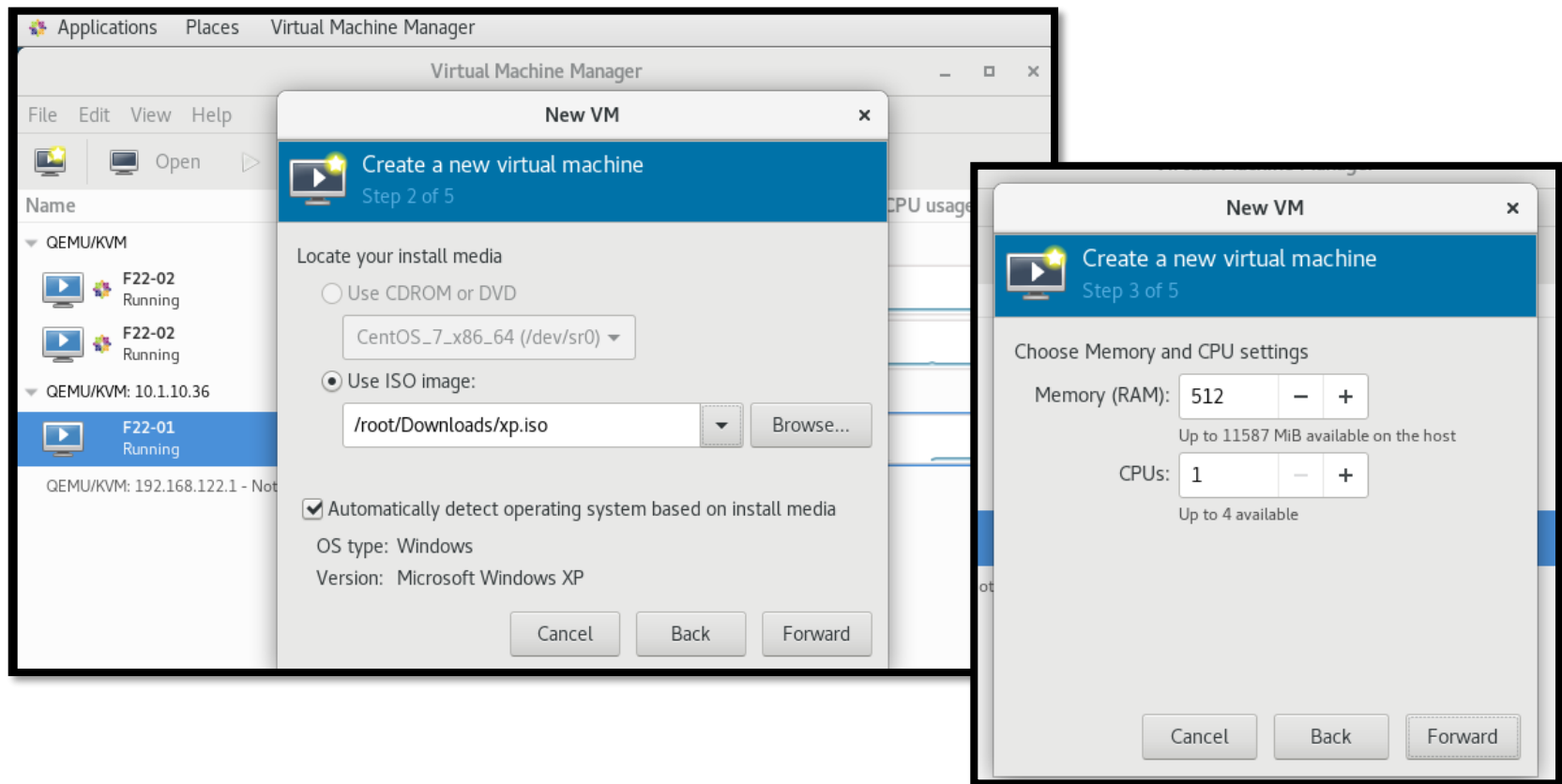




# KVM Virt-Manager(가상화 관리도구)

## 2) KVM virt manager

- virt manager를 이용하여 VM(Windows XP) 생성
  - XP이미지가 있는 경로를 지정하고 Memory(RAM)과 CPU수를 지정해준다.

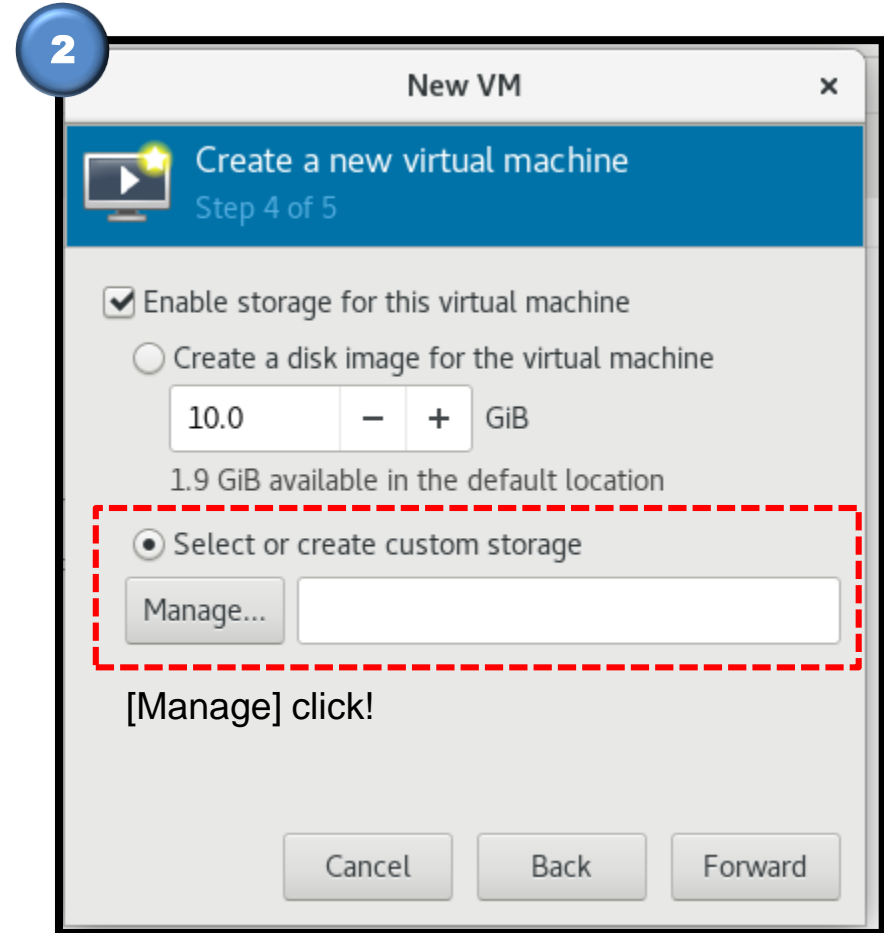
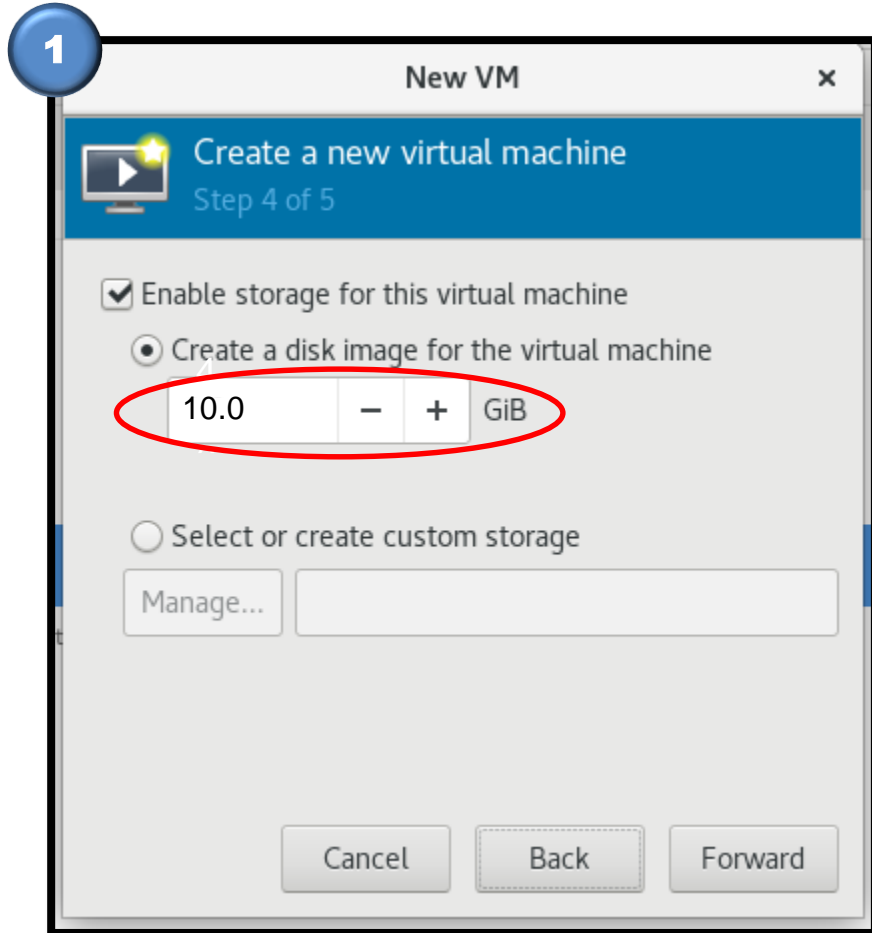


# KVM Virt-Manager(가상화 관리도구)

## 2) KVM virt manager

- virt manager를 이용하여 VM(Windows XP) 생성

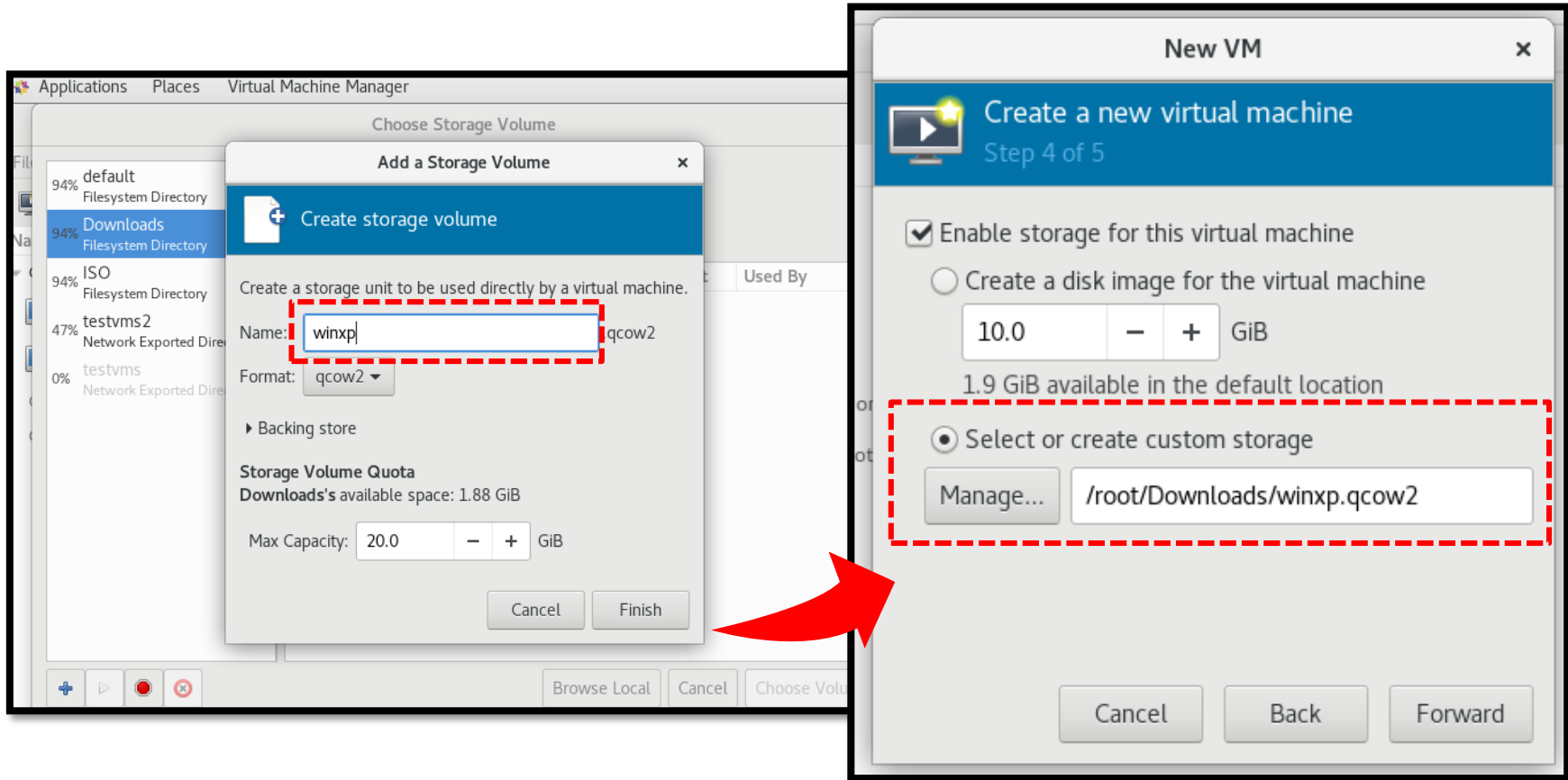
➤ 하드디스크의 크기를 설정하고서 storage에 xp전용 볼륨 만들어주기



# KVM Virt-Manager(가상화 관리도구)

## 2) KVM virt manager

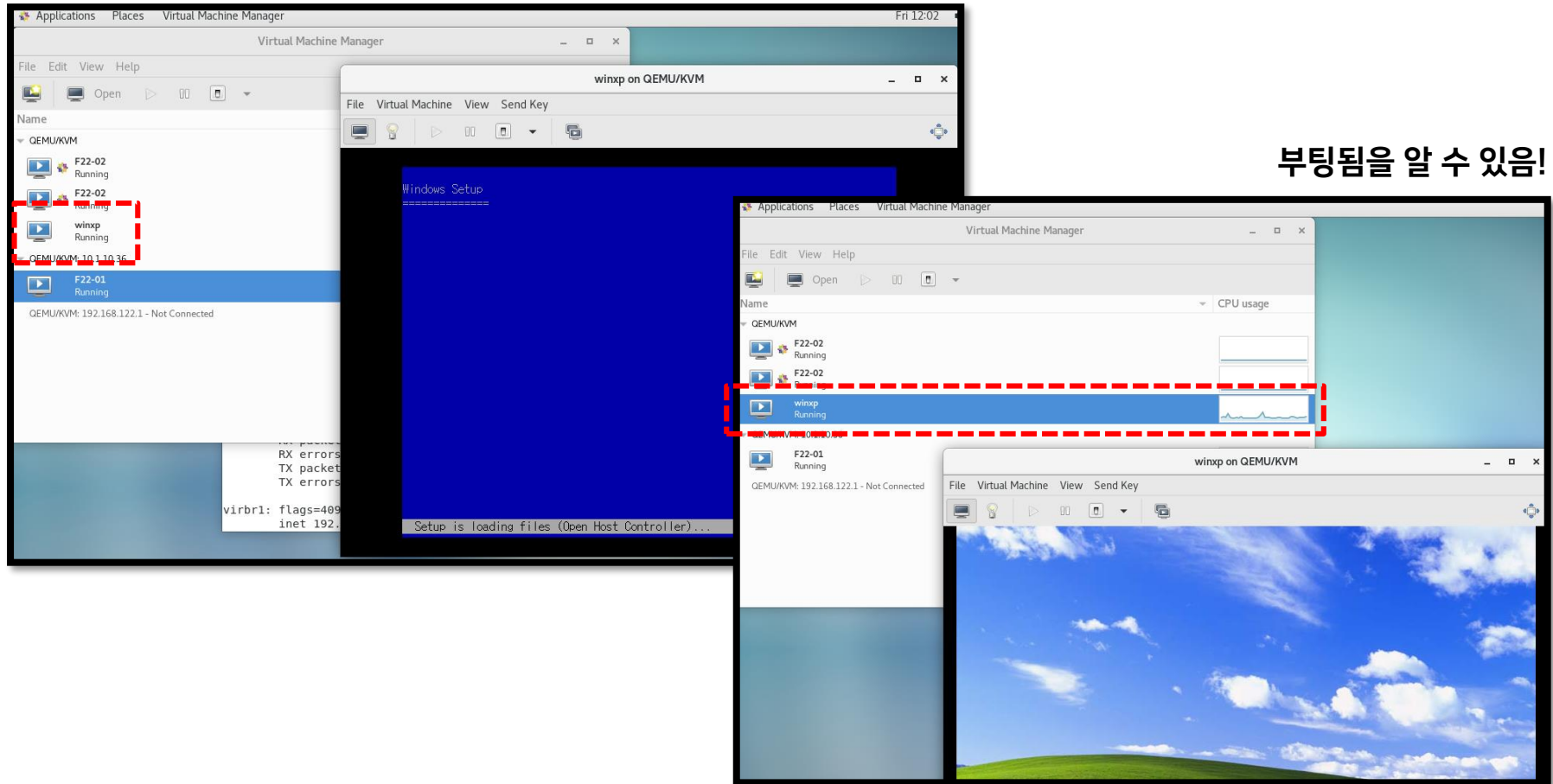
- virt manager를 이용하여 VM(Windows XP) 생성
  - 하드디스크의 크기를 설정하고서 storage에 xp전용 볼륨 만들어주기



# KVM Virt-Manager(가상화 관리도구)

## 2) KVM virt manager

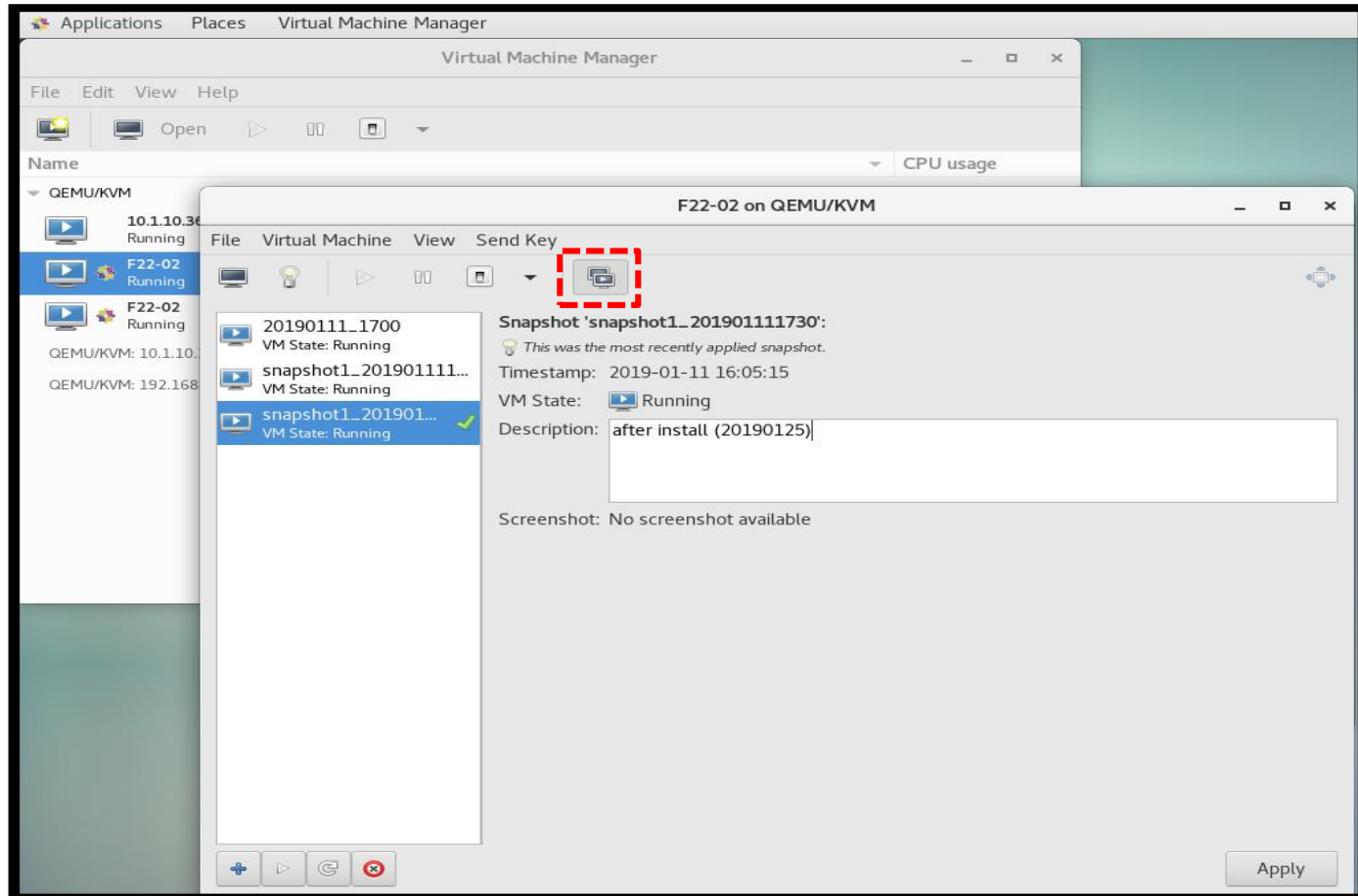
- virt manager를 이용하여 VM(Windows XP) 생성 확인
  - 생성 다 완료 후에 해당 VM에 들어가서 확인해보기.



# KVM Virt-Manager(가상화 관리도구)

## 2) KVM virt manager

- virt manager를 이용하여 Snapshot 관리하기
  - Snapshot을 이용하여 시스템복원 기능을 사용한다.

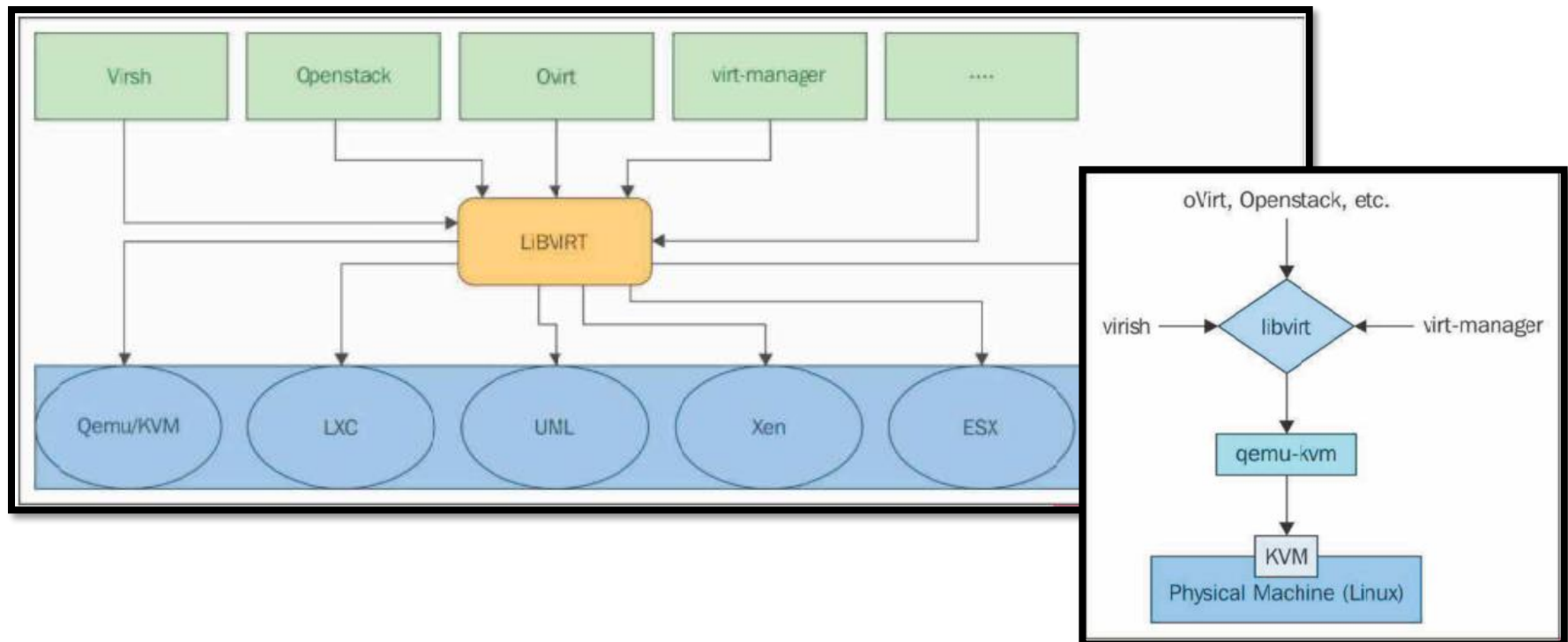


## 4. KVM Libvirt를 이용한 가상 네트워크 구축

# KVM Libvirt 서비스 구현

## KVM Libvirt란?

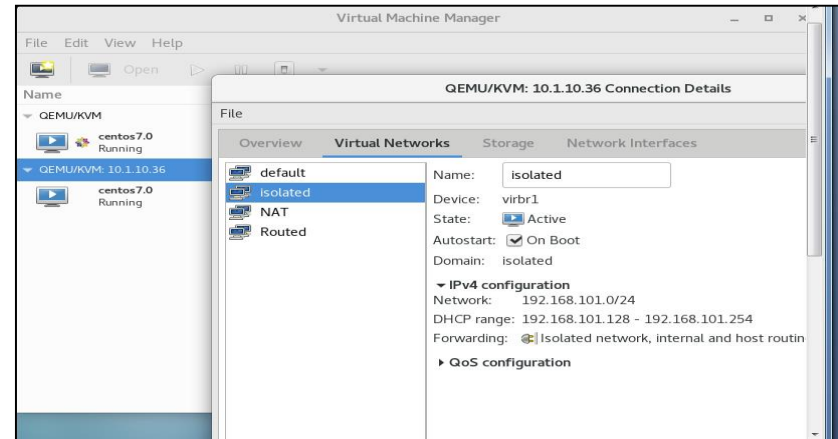
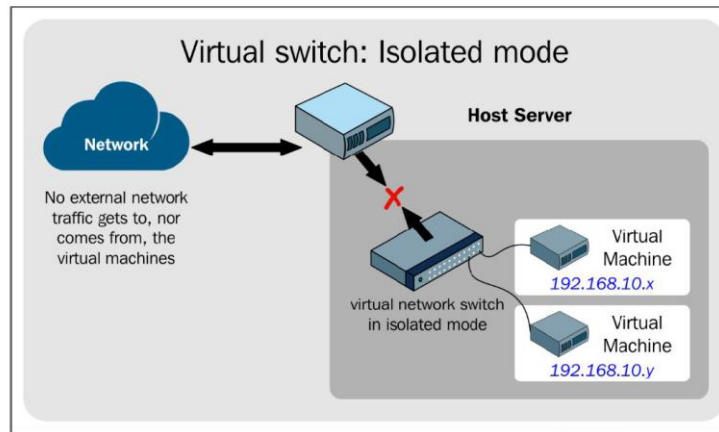
- Libvirt는 다양한 하이퍼바이저와 통신이 가능하다. 그 중 KVM/QEMU와도 통신이 가능하며 애플리케이션 프로그래밍 인터페이스(API)로 virsh(커맨드라인 인터페이스 클라이언트)와 virt-manager(그래픽유저 인터페이스 클라이언트)로 구성되어 있다.
- Libvirt는 리눅스의 libvirtd 데몬으로 실행하며 가상머신을 관리하는 관리층이라고 보면 된다. 필자는 Libvirt를 통해 CentOS를 KVM 위에 설치할 것이다.



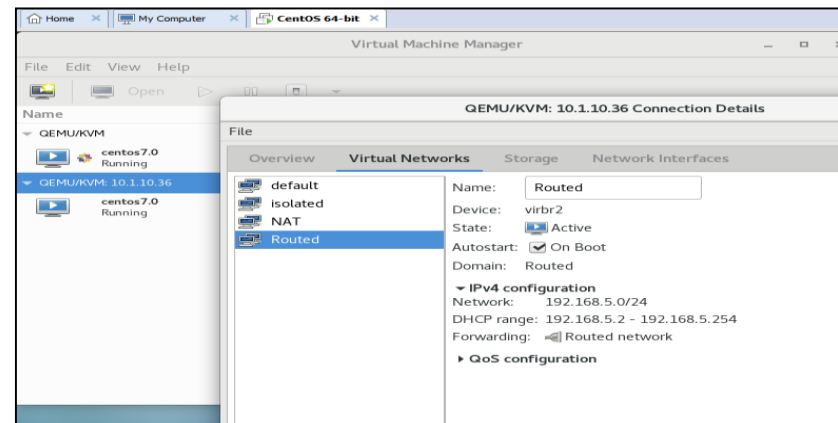
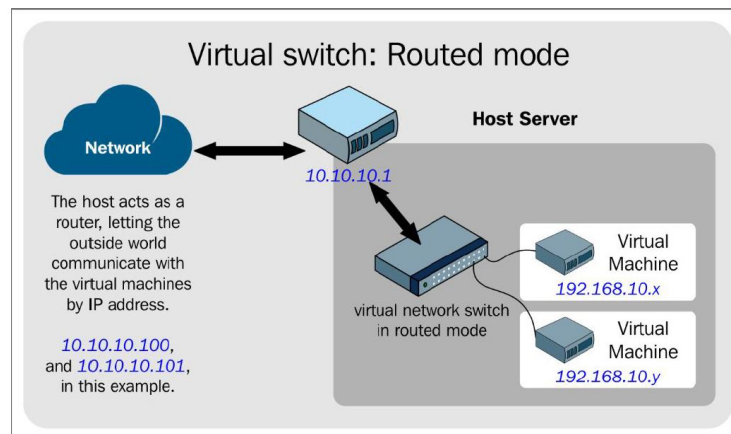
# KVM libvirt(가상 네트워킹/스토리지 생성)

## 1) Isolated 가상 네트워크 / routed 가상 네트워크

### Isolated 가상 네트워크



### Routed 가상 네트워크

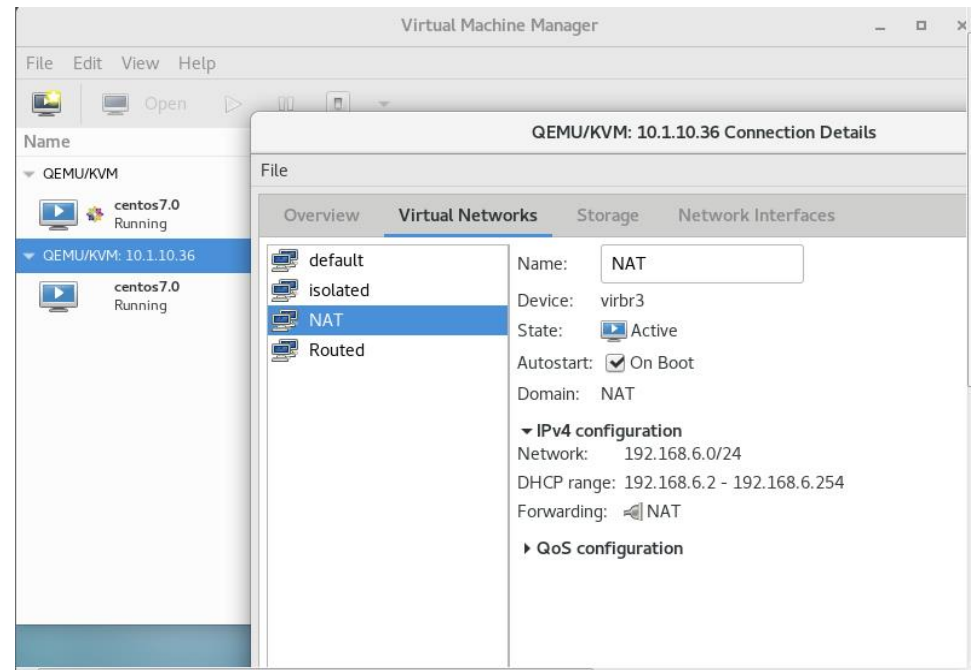
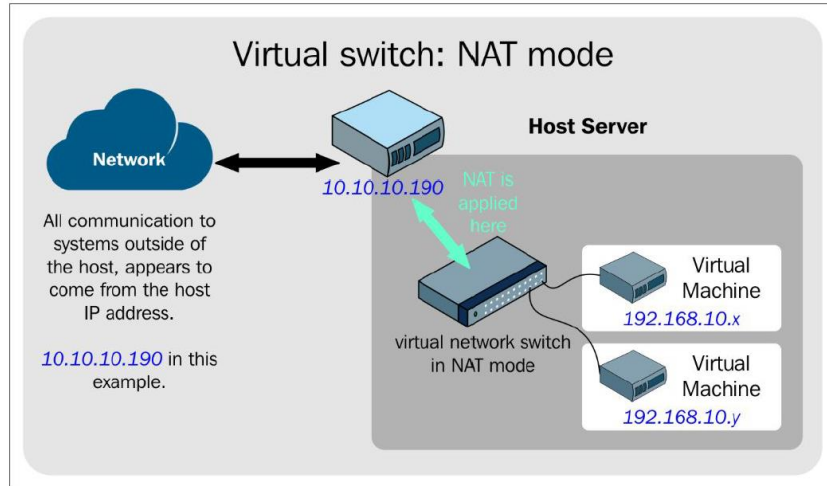




# KVM libvirt(가상 네트워킹/스토리지 생성)

## 2) NAT 가상 네트워크 생성

### NAT 가상 네트워크



# KVM libvirt(가상 네트워킹/스토리지 생성)

## 3) iSCSI 기반 스토리지 풀 생성

- 스토리지 풀 이름 입력 (iSCSI-test) 후에 Type은 logical : iSCSI Target 선택한다.

➤ iSCSI 서버 IP : 10.1.20.13

The screenshot shows a window titled "Add a New Storage Pool". Inside, there's a blue header bar with a document icon, the text "Create storage pool", and "Step 1 of 2". Below this, a message says "Select the storage pool type you would like to configure." There are two input fields: "Name:" with the value "iSCSI-test" and "Type:" with a dropdown menu showing "iscsi: iSCSI Target". A red dashed rectangle encloses these two fields. At the bottom of the window are three buttons: "Cancel", "Back", and "Forward".

# KVM libvirt(가상 네트워킹/스토리지 생성)

## 3) iSCSI 기반 스토리지 풀 생성

- iSCSI 서버 IP : 10.1.20.13

```
1) [root@localhost ~]#virsh pool-define-as --name iSCSI_test --type iscsi --source-host 10.1.20.13  
--source-dev iqn.2008-08.com.starwindsoftware:10.1.20.13-iscsi-c-2 --target /dev/disk/by-path
```

=>"Pool iSCSI\_test defined" 메시지 확인

```
2) [root@localhost ~]# virsh pool-list --all
```

Name	State	Autostart
default	active	yes
Downloads	active	yes
<b>iSCSI_test</b>	<b>inactive</b>	<b>no</b> (아직 active 안됨)
ISO	active	yes

```
3) [root@localhost ~]# virsh pool-start iSCSI_test
```

=>"Pool iSCSI\_test started" 메시지 확인

```
4) [root@localhost ~]#virsh pool-autostart iSCSI_test
```

⇒"Pool iSCSI\_test marked as autostarted" 메시지 확인

# KVM libvirt(가상 네트워킹/스토리지 생성)

## 3) iSCSI 기반 스토리지 풀 생성

- iSCSI 서버 IP : 10.1.20.13

5) [root@localhost ~]# virsh pool-info iSCSI\_test

```
virsh pool-info iSCSI_test
Name:          iSCSI_test
UUID:          eba59827-b382-4f71-90ce-1633788075f3
State:         running
Persistent:    yes
Autostart:     yes
Capacity:      300.00 GiB
Allocation:    300.00 GiB
Available:     0.00 B
```

6) [root@localhost ~]# cd /dev/disk/by-path  
[root@localhost by-path]# ll

```
lrwxrwxrwx 1 root root 9 Jan 25 14:41 ip-10.1.20.13:3260-iscsi-iqn.2008-08.com.starwindsoftware:10.1.20.13-iscsi-c-2-lun-0 -> ../../sdd
lrwxrwxrwx 1 root root 10 Jan 25 14:41 ip-10.1.20.13:3260-iscsi-iqn.2008-08.com.starwindsoftware:10.1.20.13-iscsi-c-2-lun-0-part1 -> ../../sdd1
lrwxrwxrwx 1 root root 9 Jan 25 13:56 pci-0000:00:07.1-ata-2.0 -> ../../sr0
lrwxrwxrwx 1 root root 9 Jan 25 13:56 pci-0000:00:10.0-scsi-0:0:0:0 -> ../../sda
lrwxrwxrwx 1 root root 10 Jan 25 13:56 pci-0000:00:10.0-scsi-0:0:0:0-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 Jan 25 13:56 pci-0000:00:10.0-scsi-0:0:0:0-part2 -> ../../sda2
lrwxrwxrwx 1 root root 9 Jan 25 13:56 pci-0000:00:10.0-scsi-0:0:1:0 -> ../../sdc3
```

=> 동기화되어 전송받은 파일리스트를 확인

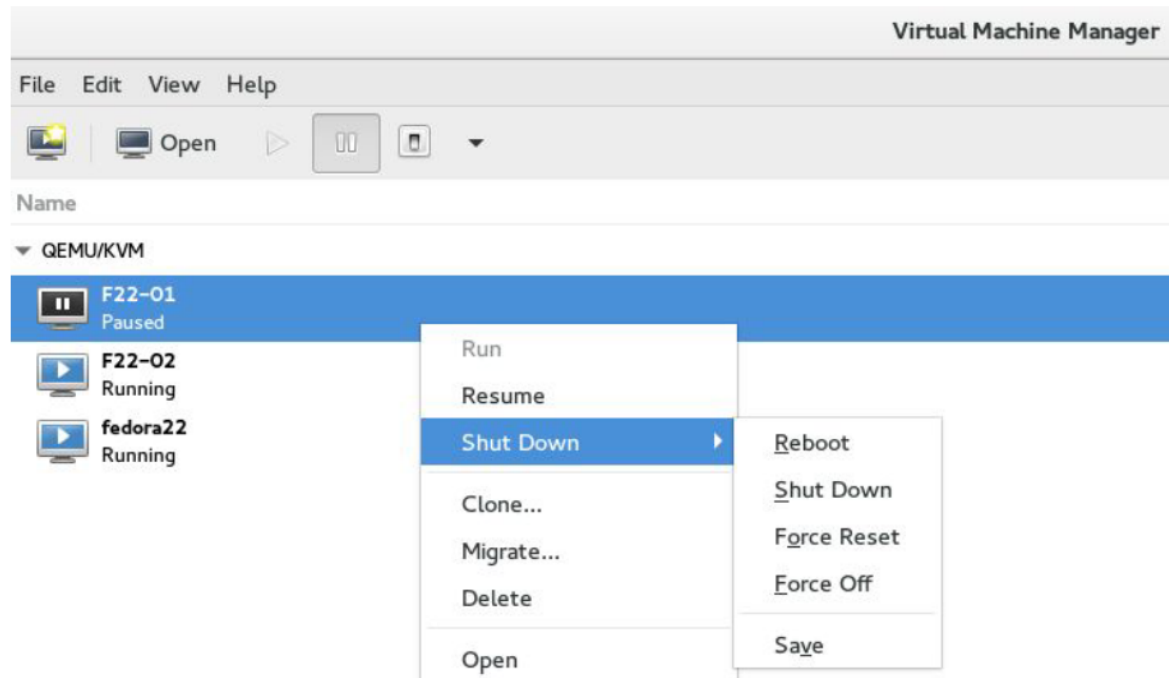
# KVM libvirt(가상 네트워킹/스토리지 생성)

## 4) 가상머신 Life Cycle 관리

- 가상머신을 리부트, shutdown, 강제리셋, 강제종료 할 수 있음.

### VM lifecycle 상태 관리

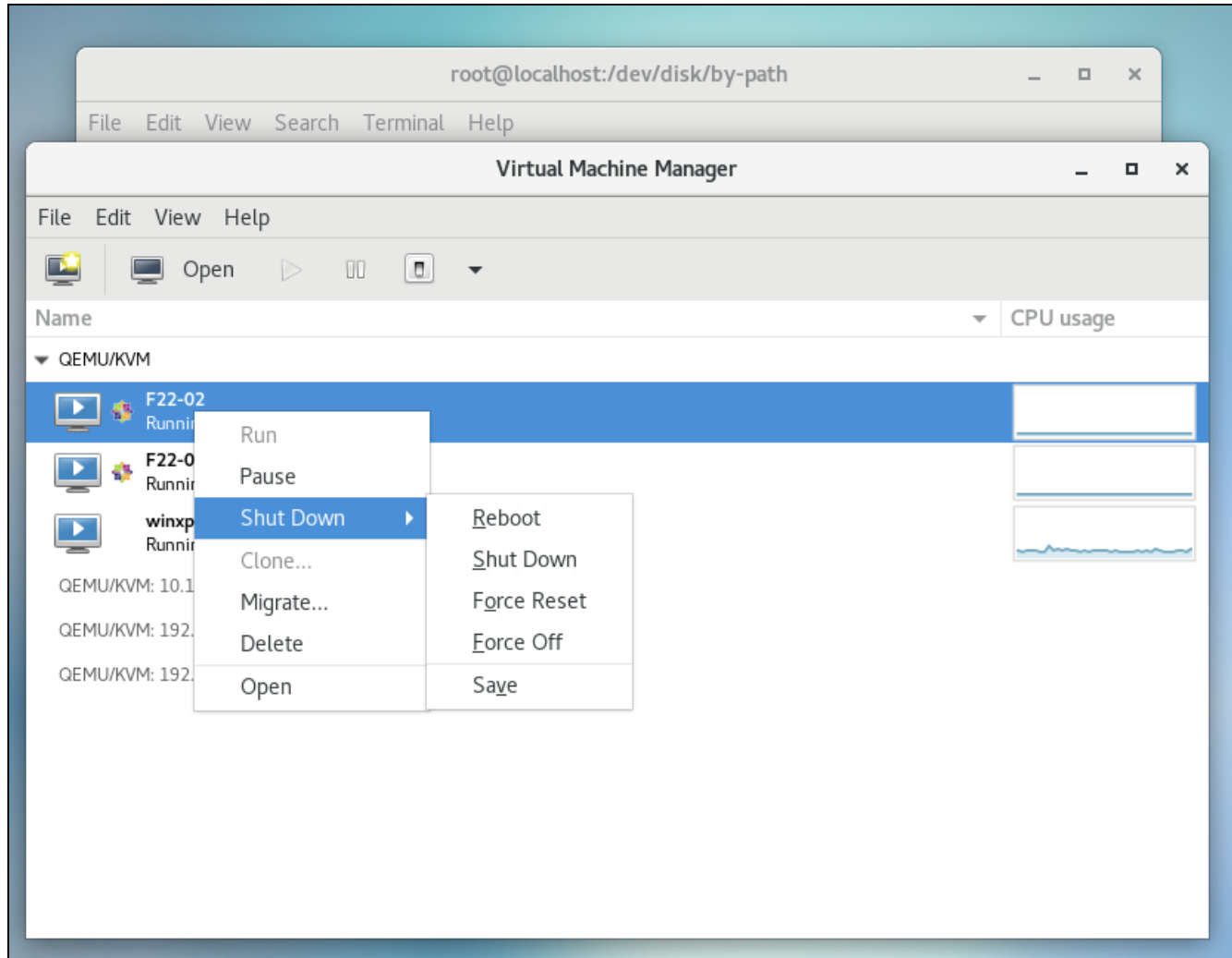
- Virt-manager 를 이용한 VM 상태 변경
  - VM 리스트 오른쪽 마우스 클릭 : VM 상태 변경 메뉴 확인



#### 4.libvirt를 사용한 가상 네트워킹/스토리지 생성

##### 5) 가상머신 Life Cycle 관리

=> 가상머신을 리부트, shutdown, 강제리셋,강제종료 함



## 5. VM 콘솔을 이용한 원격 접속

# VM 콘솔을 이용한 원격 접속

## 1) Virt-Viewer 를 이용한 VM 원격 Host 콘솔 접속

- ssh 접속 설정 후에 해당 vm으로 들어가서 확인하기

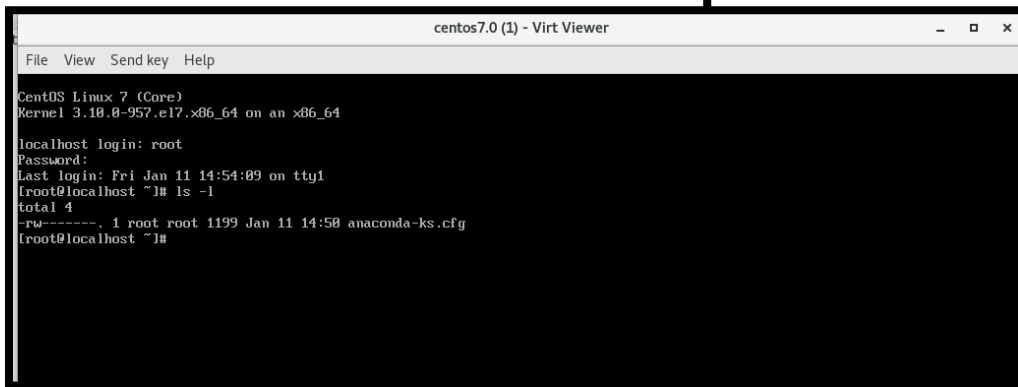
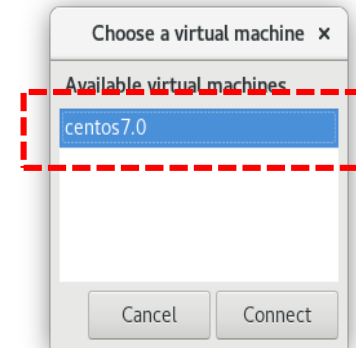
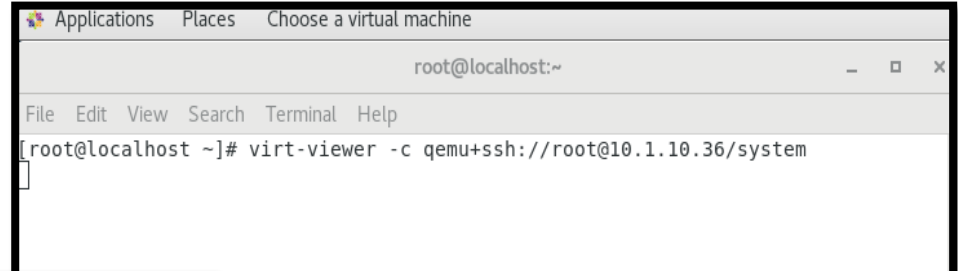
- Virt-view 를 이용한 그래픽 콘솔 접속

- Ssh 접속 설정

```
# ssh-keygen  
# ssh-copy-id root@192.168.122.1
```

- Ssh 접속 설정

```
# virt-viewer -c qemu+ssh://root@192.168.122.1/system
```



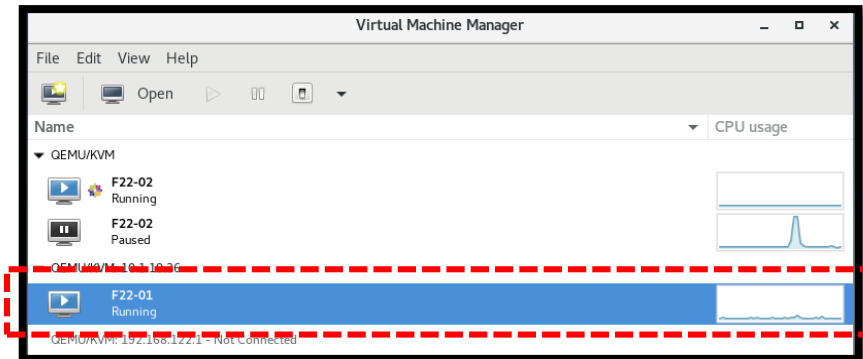
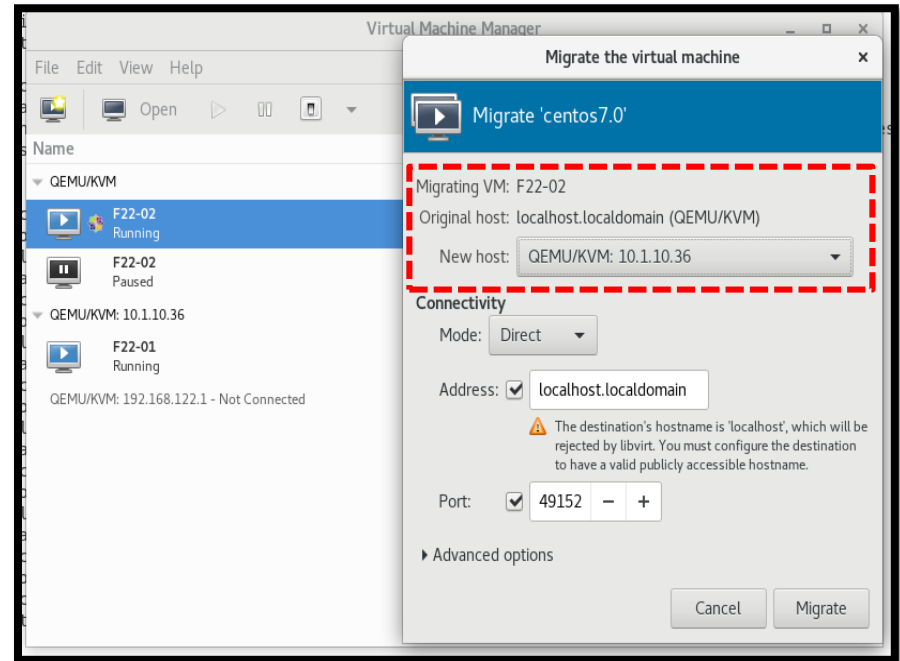
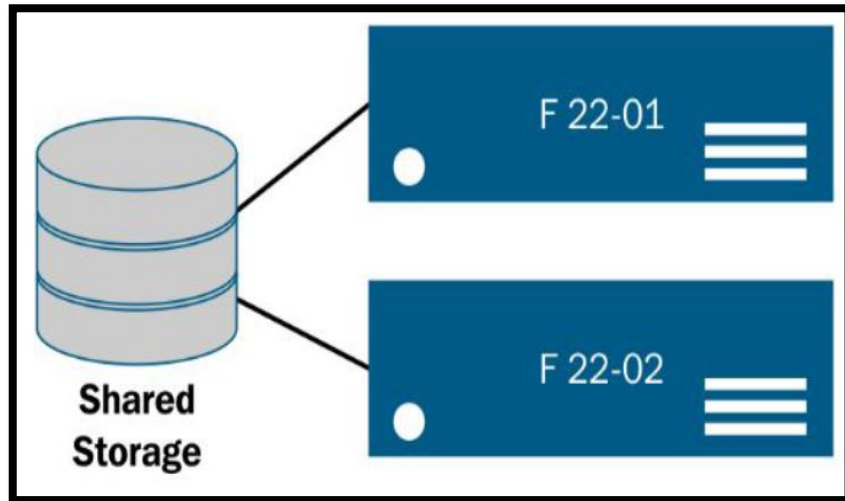


## 6. NFS를 접속을 통한 VM 마이그레이션

# NFS 접속을 통한 VM 마이그레이션

## 1) VM 마이그레이션을 위한 스토리지 구성 및 마이그레이션

- Offline/ live 마이그레이션 수행을 위한 공유 스토리지 구성 및 마이그레이션
  - VM은 두대의 KVM 하이퍼바이저의 공유 스토리지에서 운영중(F22-01, F22-02)
  - NFS를 통해 F22-02의 VM을 F22-01 (10.1.10.36)로 마이그레이션 한다.



또 다른 KVM에서 새롭게 만든 VM

# NFS 접속을 통한 VM 마이그레이션

## 2) VM 마이그레이션 설정 및 진행하기

- NFS를 통한 KVM 마이그레이션 실행하기

### [VM 마이그레이션 설정]

```
echo '/testvms *(rw,sync,no_root_squash)' >> /etc/export
```

```
systemctl start rpcbind nfs-server
```

```
systemctl enable rpcbind nfs-server
```

```
showmount -e
```

```
mkdir -p /var/lib/libvirt/images/testvms/
```

```
virsh pool-define-as --name testvms --type netfs --source-host 10.1.10.47 --source-path /testvms  
--target /var/lib/libvirt/images/testvms/
```

=> F22-01(호스트 VM, IP : 10.1.10.36 에서 설정)

```
virsh pool-define-as --name testvms2 --type netfs --source-host 10.1.10.36 --source-path /share/  
--target /var/lib/libvirt/images/testvms/
```

=> F22-02(호스트 VM, IP : 10.1.10.47 에서 설정)

```
mount 10.1.10.36:/share /mnt
```

# NFS 접속을 통한 VM 마이그레이션

## 3) VM 마이그레이션 결과 확인하기

- 제대로 마이그레이션이 되었는지 확인하기
- 22-02의 내용이 22-01의 내용으로 마이그레이션되었는지 확인하기.

[FS22-01의 공유폴더 내용]

```
[root@localhost images]# ls -l /share
```

```
total 16064
```

```
-rwxr-xr-x 1 root root 4902656 Jan 17 11:22 vmlinuz-0-rescue-90bc9bd3d94243aea1a172a3bd599264  
-rwxr-xr-x 1 root root 4902656 Jan 17 11:22 vmlinuz-3.10.0-123.el7.x86_64  
-rwxr-xr-x 1 root root 6639808 Jan 17 11:22 vmlinuz-3.10.0-957.1.3.el7.x86_64
```

[FS22-02의 공유폴더 내용]

```
[root@localhost images]# ls -l /var/lib/libvirt/images/testvms/
```

```
total 16064
```

```
-rwxr-xr-x 1 root root 4902656 Jan 17 11:26 vmlinuz-0-rescue-8a238520050e4aaf8666989bec503213  
-rwxr-xr-x 1 root root 4902656 Jan 17 11:26 vmlinuz-3.10.0-123.el7.x86_64  
-rwxr-xr-x 1 root root 6639808 Jan 17 11:26 vmlinuz-3.10.0-957.1.3.el7.x86_64
```

- 같은 내용으로 똑같이 있기에 마이그레이션이 잘 되었음.

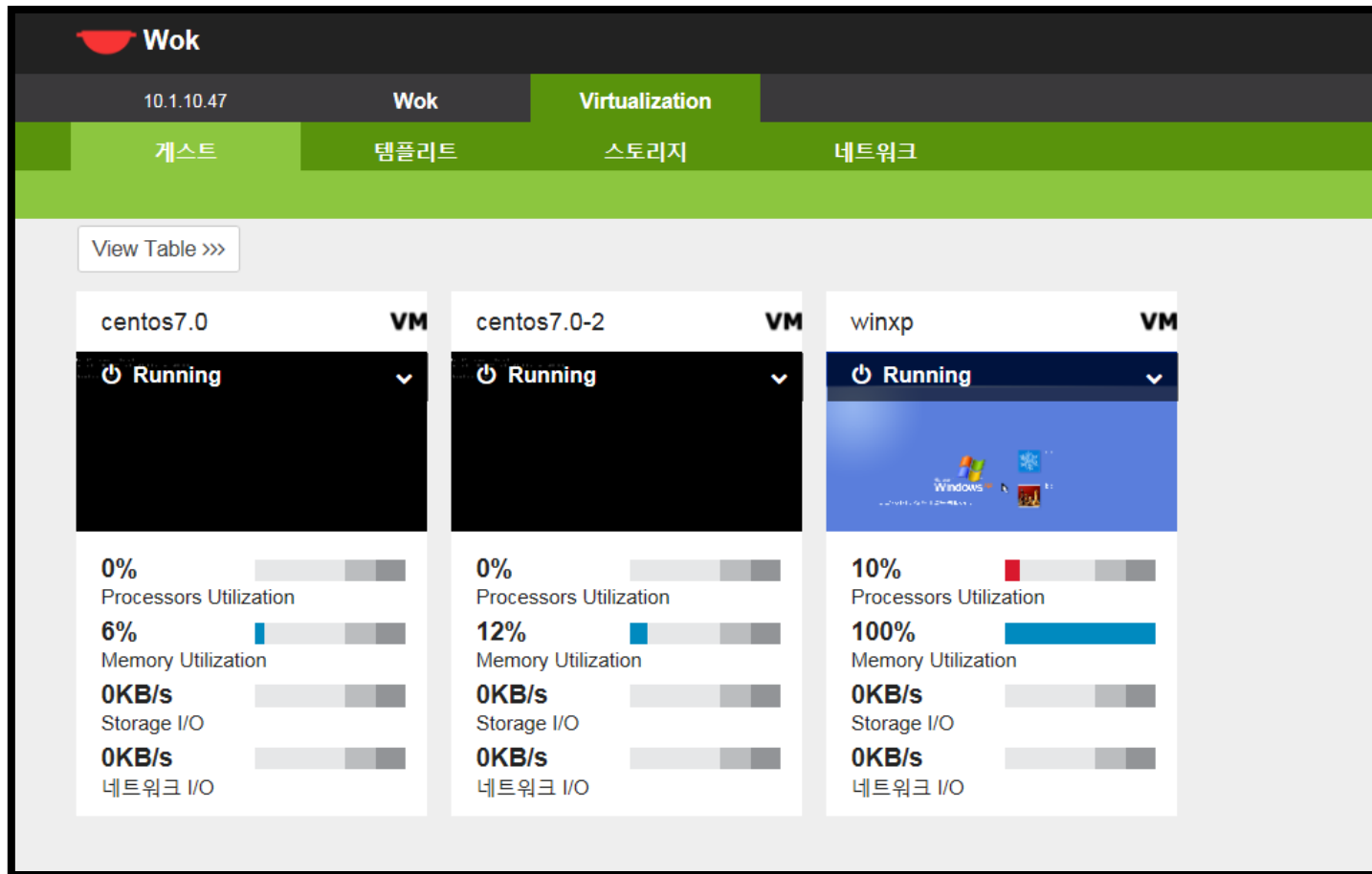
## 7. HTML5기반 Kvm/Libvirt 웹관리도구 “Kimchi” 구축

# Kimchi 웹 프로그램

## 1) Kimchi 웹 프로그램 이용하기

- HTML5기반 KVM/libvirt 관리 도구로 VM의 CPU, Memory 상태를 실시간 점검할 수 있다.

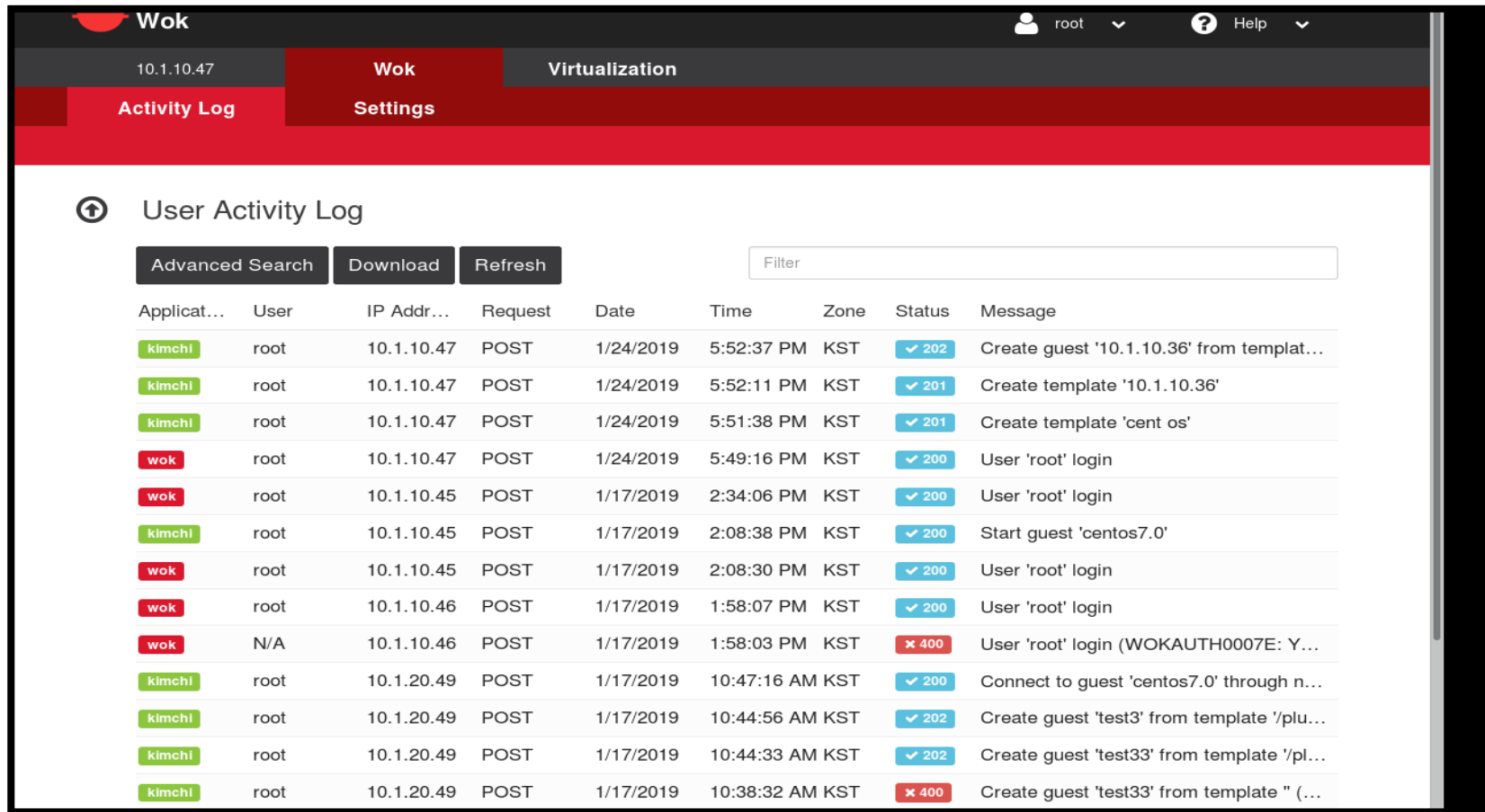
➢ Kvm 내에서 VM이 구동될 경우 바로 알람이 뜬다!



# Kimchi 웹 프로그램

## 1) Kimchi 웹 프로그램 이용하기

- HTML5기반 KVM/libvirt 관리 도구로 VM의 실시간 로그 파일도 확인 가능하다.



The screenshot displays the Wok web interface. At the top, there's a navigation bar with 'Wok' and 'Virtualization' tabs. Below this, a red bar contains 'Activity Log' and 'Settings' links. The main content area is titled 'User Activity Log' and includes buttons for 'Advanced Search', 'Download', and 'Refresh', along with a 'Filter' input field. A table lists various activities with columns for Application, User, IP Address, Request, Date, Time, Zone, Status, and Message.

Applicat...	User	IP Addr...	Request	Date	Time	Zone	Status	Message
kimchi	root	10.1.10.47	POST	1/24/2019	5:52:37 PM	KST	✓ 202	Create guest '10.1.10.36' from templat...
kimchi	root	10.1.10.47	POST	1/24/2019	5:52:11 PM	KST	✓ 201	Create template '10.1.10.36'
kimchi	root	10.1.10.47	POST	1/24/2019	5:51:38 PM	KST	✓ 201	Create template 'cent os'
wok	root	10.1.10.47	POST	1/24/2019	5:49:16 PM	KST	✓ 200	User 'root' login
wok	root	10.1.10.45	POST	1/17/2019	2:34:06 PM	KST	✓ 200	User 'root' login
kimchi	root	10.1.10.45	POST	1/17/2019	2:08:38 PM	KST	✓ 200	Start guest 'centos7.0'
wok	root	10.1.10.45	POST	1/17/2019	2:08:30 PM	KST	✓ 200	User 'root' login
wok	root	10.1.10.46	POST	1/17/2019	1:58:07 PM	KST	✓ 200	User 'root' login
wok	N/A	10.1.10.46	POST	1/17/2019	1:58:03 PM	KST	✗ 400	User 'root' login (WOKAUTH0007E: Y...
kimchi	root	10.1.20.49	POST	1/17/2019	10:47:16 AM	KST	✓ 200	Connect to guest 'centos7.0' through n...
kimchi	root	10.1.20.49	POST	1/17/2019	10:44:56 AM	KST	✓ 202	Create guest 'test3' from template '/plu...
kimchi	root	10.1.20.49	POST	1/17/2019	10:44:33 AM	KST	✓ 202	Create guest 'test33' from template '/pl...
kimchi	root	10.1.20.49	POST	1/17/2019	10:38:32 AM	KST	✗ 400	Create guest 'test33' from template " (...

# Kimchi 웹 프로그램

## 1) Kimchi 웹 프로그램 이용하기

- 가상 네트워크 생성 및 조회 기능/ 스토리지 생성 및 조회 기능이 있다

The top screenshot displays the 'Network' tab in the Kimchi web interface. It features a table with columns: 네트워크 이름 (Network Name), 네트워크 유형 (Network Type), 인터페이스 (Interface), and Address Space. The table lists six networks: NAT, Routed, default, isolated, isolated, and ttt. Each row has a status icon (green power button) and an action button (조치) with a dropdown arrow. A 'Filter' input field is located at the top right of the table.

네트워크 이름	네트워크 유형	인터페이스	Address Space
NAT	nat	virbr3	192.168.15.0/24
Routed	route	virbr2	192.168.5.0/24
default	nat	virbr0	192.168.122.0/24
isolated	isolated	virbr4	사용 불가능
isolated	isolated	virbr1	192.168.101.0/24
ttt	bridge	kbeno33554960	사용 불가능

The bottom screenshot displays the 'Storage' tab in the Kimchi web interface. It features a table with columns: 이름 (Name), 유형 (Type), Location, %Used, Allocated, Capacity, and Disks. The table lists six storage volumes: Downloads, ISO, default, iSCSI\_test, testvms, and testvms2. Each row has a status icon (green power button) and an action button (조치) with a dropdown arrow.

이름	유형	Location	%Used	Allocated	Capacity	Disks
Downloads	dir	/root/Downloads	95%	33.6G	35.5G	▼
ISO	dir	/var/lib/kimchi/isos	98%	34.9G	35.5G	▼
default	dir	/var/lib/libvirt/images	95%	33.6G	35.5G	▼
iSCSI_test	iscsi	/dev/disk/by-path	100%	300.0G	300.0G	▼
testvms	netfs	/var/lib/libvirt/image...	0%	0.0B	0.0B	▼
testvms2	netfs	/var/lib/libvirt/image...	48%	17.0G	35.5G	▼



# Kimchi 웹 프로그램

## 2) Kimchi 웹 프로그램 이용하여 Template 작성하기

- 템플릿 추가를 통해서 나만의 템플릿을 만들어 추가할 수 있다.

The image shows a screenshot of the Kimchi web interface. On the left, a sidebar menu is visible with a green button labeled '템플릿 추가' (Add Template) highlighted by a red dashed box. A red arrow points from this button to the 'Add Template' dialog box on the right. The dialog box is titled '템플릿 추가' (Add Template) and contains the following fields and options:

- Template Name:** A text input field with 'Windows XP' entered.
- Where is the source media for this template?:** Radio buttons for 'Image Template' (selected) and 'Netboot Template'.
- 파일 경로:** A text input field with '/root/Downloads/xp.iso' entered.
- Buttons:** '작성' (Save) and '취소' (Cancel) buttons at the bottom.

A red error message is displayed at the bottom of the dialog box: 'The server encountered an unexpected condition which prevented it from fulfilling the request.'

# Kimchi 웹 프로그램

## 2) Kimchi 웹 프로그램 이용하여 Template 확인

- 이전 페이지에서 만든 템플리트(Windows XP) 확인하기

➤ Virtualization > 템플리트 > Windows XP 가 있음 확인하기

The screenshot shows the Wok Virtualization management interface. The top navigation bar includes 'Wok' and a user profile 'root'. Below this, a secondary navigation bar has tabs for '게스트', '템플리트', '스토리지', and '네트워크'. The '템플리트' tab is selected. On the right, there is a '+ 템플리트 추가' button. Below the navigation bar, there is a 'View Gallery >>>' button and a 'Status' dropdown menu. The main content area displays a table of templates with the following columns: Template Name (ID), OS, 버전, Current CPUs, and 메모리. The table contains three entries: 1. Template Name (ID): 10.1.10.36, OS: centos, 버전: 7, Current CPUs: 1 코어, 메모리: 1024 M. 2. Template Name (ID): Windows XP, OS: VM unknown, 버전: unknown, Current CPUs: 1 코어, 메모리: 1024 M. This row is highlighted with a red dashed box. 3. Template Name (ID): cent os, OS: centos, 버전: 7, Current CPUs: 1 코어, 메모리: 1024 M. To the right of each row is a '조치' (Action) button with a dropdown arrow.

Template Name (ID)	OS	버전	Current CPUs	메모리	조치
10.1.10.36	centos	7	1 코어	1024 M	조치 ▼
Windows XP	VM unknown	unknown	1 코어	1024 M	조치 ▼
cent os	centos	7	1 코어	1024 M	조치 ▼

## 느낀점

본 프로젝트는 기존에 수행하였던 방식과 다르게, 스위치 이용하여 망 구성을 하고, KVM을 설치한 후 구동시켜 진행 하였습니다. 구성 방식을 세부적으로 설명하면, Switch 3 대를 이용하여 네트워크 망을 구성한 후, USB를 사용하여 Bare metal PC에 ESXi와 CentOS 7를 기반으로 KVM Virt-Manager, VM콘솔, VM 마이그레이션, 그리고 HTML5 기반 KVM/libvirt 관리도구인 “김치” 웹프로그램을 구축 및 운용하였습니다.

해당 프로젝트 진행 전에 미리 VMware Workstation 등을 이용하여 실습을 진행했었던 덕분에, 이번 프로젝트 진행시 특별한 애로사항은 없었습니다. 하지만 망구성 시에 IP 분배 라던지, ESXi의 랜카드 인식 실패 등, 그 전 실습에서는 겪어보지 못했던 현장(비록 완벽한 현장은 아니었지만) 상황을 느껴볼 수 있었던 소중한 경험이었으며, 다른 한편으로는 아직 사용해 보지 못한 기능 및 상황들이 많고, 앞으로도 공부해야할 사항들이 많다는 것 또한 느꼈습니다. 그리고 snmp 프로토콜을 사용하는 것은, 많은 편의성 및 관리 용이성을 제공하지만, 관리에 실패한다면 부정적인 결과를 초래할 가능성이 높아진다는 걸 알게 되었습니다. 끝으로, 이번 프로젝트를 진행하면서 아쉬운 사항으로는 docker hub에서 가져온 cacti를 사용해 보았지만, 사용만 해보았을 뿐 세팅해 보지 못한 부분입니다. 그래서 앞으로는 snmp에 관해서 더 많은 사항들에 대해 알아보며, 이를 스터디하고 체득함으로써, 실무에 투입되어 직무를 수행시 다양한 케이스에 대처할 수 있는 능력을 확보할 수 있을 것이라는 생각을 하게 되었습니다.