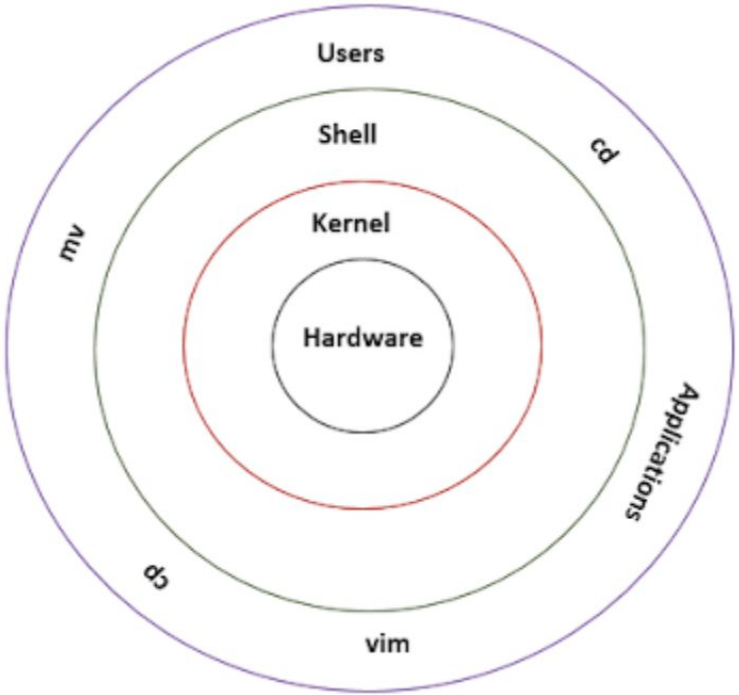
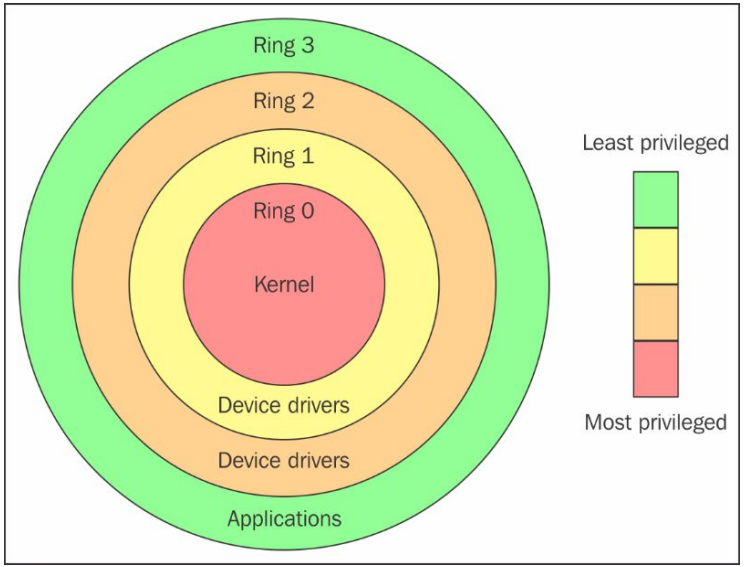
시스템프로그램 : 가상화

SHELL



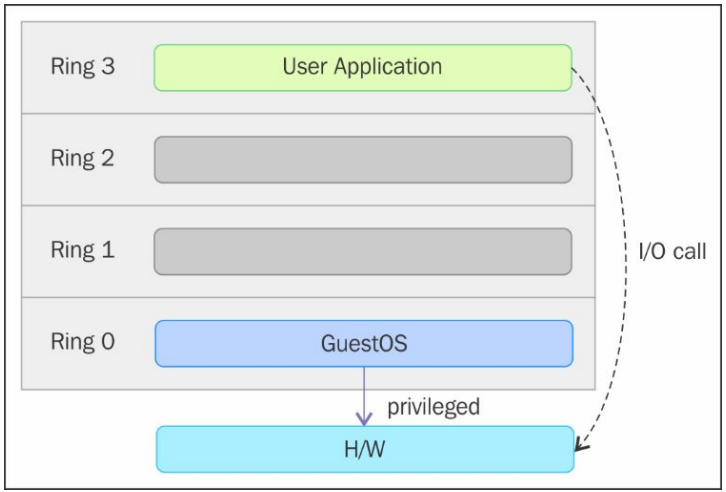
링 보호 (Protection Ring)

- 링 : 컴퓨터 시스템의 리소스 접근시 데이터 와 보안을 위한 장치
  - 링 0 : 커널모드 / 슈퍼 관리자 모드
  - 링 3 : 사용자 모드 / 애플리케이션 실행 모드

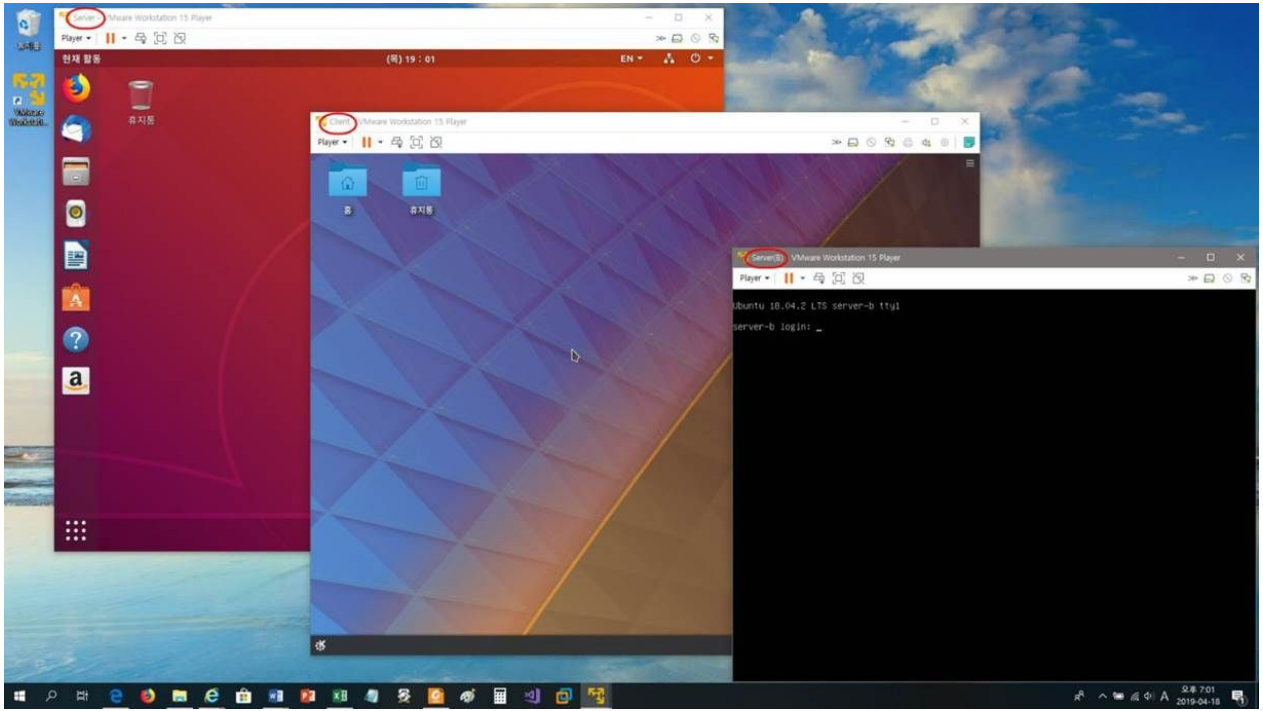


링 0 (커널모드) / 링 3 (사용자모드)

- 운영체제(리눅스/윈도우)는 커널모드 와 사용자 모드를 모두 사용
  - 사용자모드 실행은 커널 호출 없이 메모리, CPU, I/O 포트의 접근이 제한됨
  - 특별한 권한(커널모드) 을 위해 시스템 콜(System Call) 실행 필요
  - 커널모드로 실행된 작업은 사용자 모드에 그 결과를 반환
  - 운영체제는 리소스 관리나 하드웨어 접근을 위해 링 0 에 존재해야 함



가상머신


Server
VMware Workstation 15 Player
Player
현재 활동
(목) 19 : 01
EN
휴지통
Client
VMware Workstation 15 Player
Player
홈
휴지통
Server(b)
VMware Workstation 15 Player
Player
Ubuntu 18.04.2 LTS server-b tty1
server-b login:
오후 7:01
2019-04-18

하이퍼바이저 / Guest VM

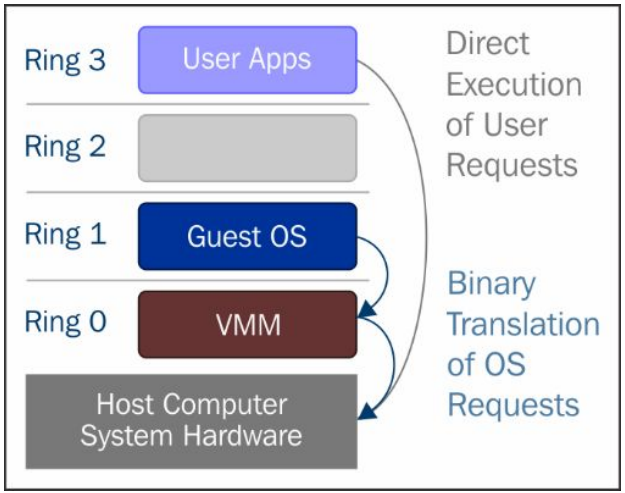
- 하이퍼바이저/가상서버 모니터는 링 0 에서 실행 됨
  - 호스트의 메모리, CPU, I/O 포트 접근이 필요함
- Guest VM 은 링 1 에서 실행 됨
  - 하이퍼바이저 와 VMM 링 0 에서 실행 되므로 Guest VM은 링 1에서 실행됨
  - Guest VM 의 OS 는 가상화 레이어(링 1)에 대해 인지 하지 못함
  - 링 0 의 권한이 필요한 자원에 접근 해야함
- 동일시간에 링 0 에서 하나의 커널 명령만 수행 가능

Guest OS 실행 링

- Guest OS 는 링 0 보다 권한이 낮은 링 1 에서 실행
- 또는 사용자 모드 링 3 에서 실행 할 수 있도록 운영체제 수정

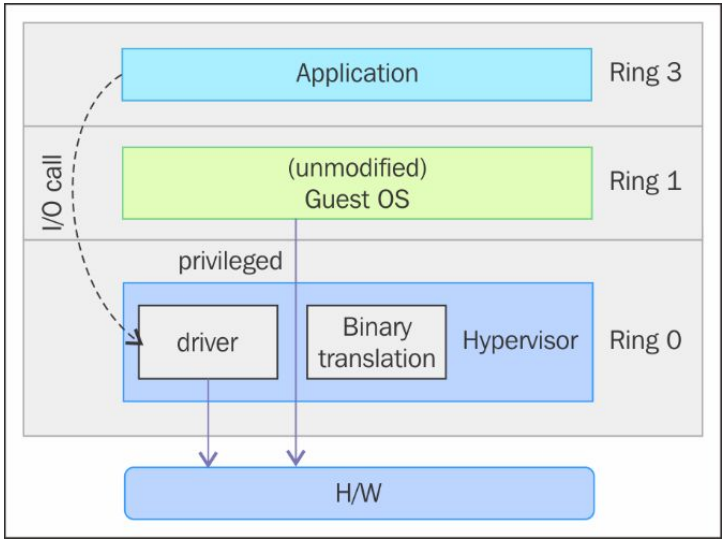
전 가상화 (Full Virtualization)

- Guest OS 는 링 1 에서 실행
  - OS 권한 제약 사항을 극복하기 위해 자원이 에뮬이션 기능 사용
- 가상서버 모니터(VMM) 은 링 0 에서 실행
- 전 가상화는 x86 아키텍처에서 처음 실행됨
- 가상화 되지 않는 명령은 바이너리 변환 기술을 이용



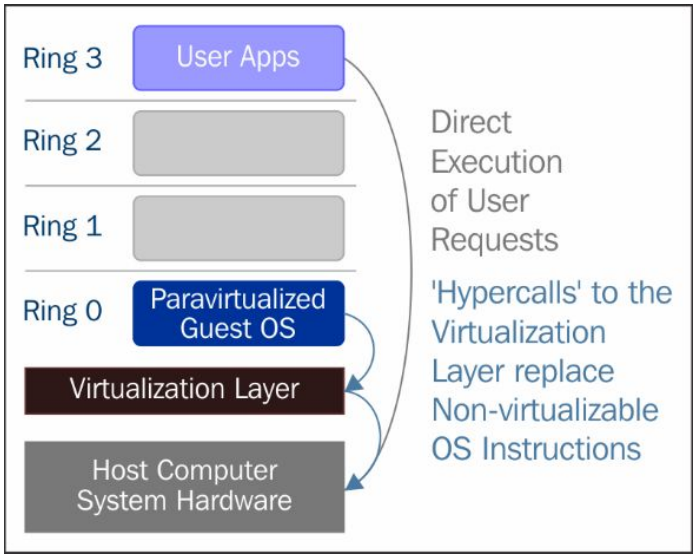
전 가상화 (Full Virtualization)

- 권한이 필요한 명령은 VMM 에 전달되어 바이너리 변환
- 바이너리 변환방식은 큰 성능 저하를 발생
- OS 의 커널 수정없이 사용 가능함
- VMM 은 CPU 를 관리하고 에뮬레이션 기능을 제공



반 가상화 (Para Virtualization)

- Guest OS 링 0 에 접근 하기 위해 운영체제가 수정되어야 함
- Guest OS는 하이퍼바이저/VMM 사이에 하이퍼콜(Hypercall) 호출 실행
- 하이퍼바이저는API를 제공
- 권한이 필요한 명령은API 를 통해 실행 되어 링 0 에서 실행됨



반 가상화 (Para Virtualization)

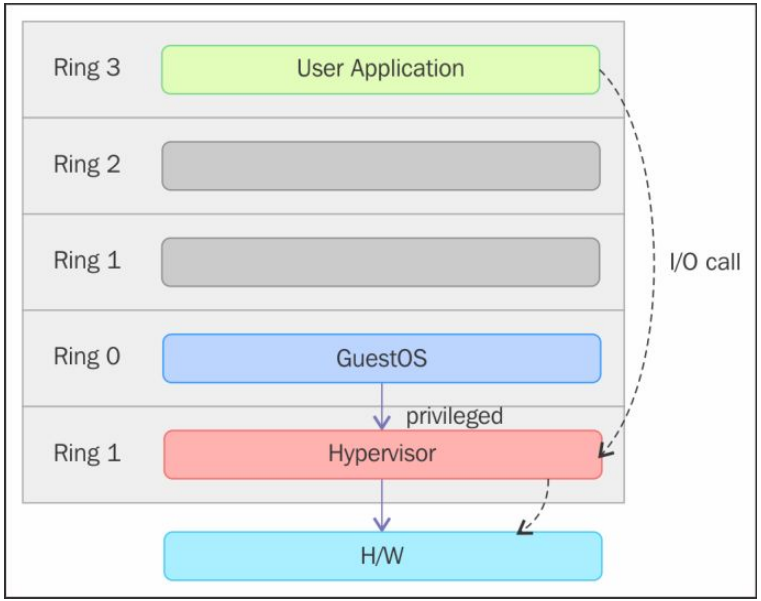
- Guest OS는 자신의 VM 이 가상화 되었음을 인지하고 있음
- 권한이 필요한 명령은 VMM 에 전달되기 위해 하이퍼콜 호출
- Guest OS 커널은 하이퍼콜을 통해 VMM 과 직접 통신 가능
- 바이너리 변환이 필요한 전가상화와 비교해 큰 성능 향상
- 반가상화 인지 가능한 특수한 Guest OS 커널이 필요

하드웨어 가상화 지원기능

- Intel 과 AMD 는 가상화가 x86 아키텍처의 중요 과제로 인식
- 프로세스 확장을 통한 독립적인 가상화 지원 기능 개발
  - Intel : Intel VT-x
  - AMD : AMD-V
  - Itanium : VT-i
- IA-32 명령어를 확장
  - Intel : VT(Virtualization Technology)
  - AMD : SVM(Secure Virtual Machine)
- 하이퍼바이저와 VMM 이 Guest OS를 Ring 0 에서 실행 가능하도록 지원
- VM 을 링 0 에서 실행 가능 할 수 있도록, 하이퍼바이저 와 VMM 을 링 - 1 에서 실행함
- 기존 전가상화 기술보다 성능이 향상됨

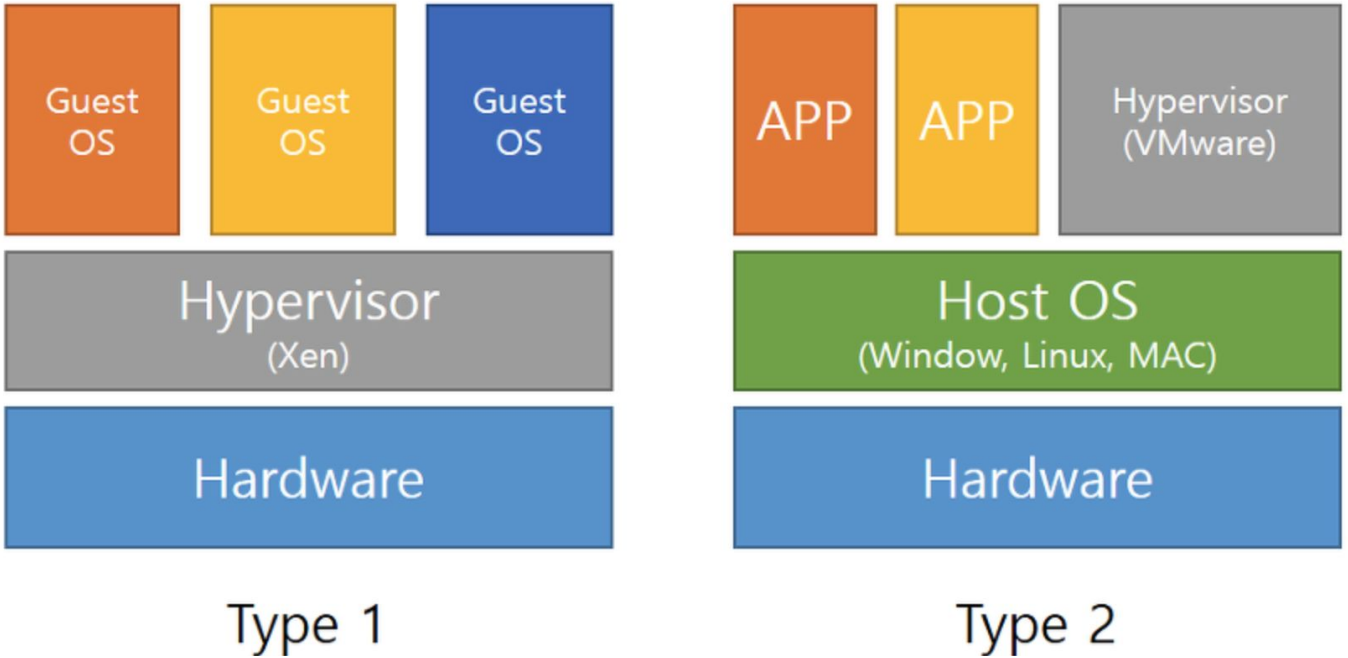
하드웨어 가상화 지원기능

- 하드웨어가 가상화를 인식
- 가상화 솔루션 개발시 복잡함을 줄여줌
- KVM 은 하드웨어 가상화 지원 기능을 사용



VM 가상화의 종류

- VM을 직접 올리는 경우



Type 1 하이퍼바이저

특징: 하드웨어에 직접 설치되어 실행

장점: 높은 성능, 보안성, 안정성

사용 사례: 기업용 서버, 데이터센터

주요 제품:

- VMware ESXi
- Microsoft Hyper-V
- Citrix XenServer
- KVM (Kernel-based Virtual Machine)

Type 1

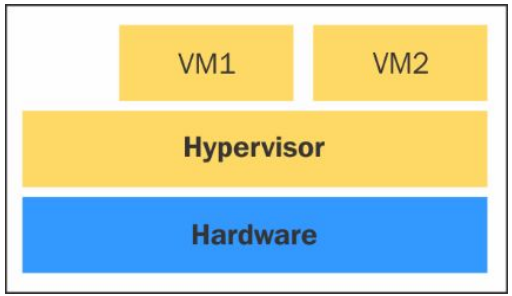
- XEN
  - 최초의 가상머신
  - 리눅스에 기본장착
  - Citrix사의 제품
  - 코드는 무료공개
- VMWare ESXi
  - VMWare사의 제품
  - 64-bit 운영체제에만 동작함
  - 코드는 절대 공개안함
  - 브라우저를 통해 외부에서 접속 가능

Type1 , Type2 하이퍼바이저

- 하이퍼바이저는 존재 위치에 따라 그 분류가 달라진다.
  - Type1 : 하드웨어 위에서 직접 실행
  - Type2 : 운영체제가 있고, 하이퍼바이저가 분리된 층에 실행
- Type1 하이퍼바이저
  - 시스템 하드웨어와 직접 상호작용
  - 운영체제가 필요하지 않음
  - 베어메탈, 임베디드, 네이티브 하이퍼바이저라고 함
- Type2 하이퍼바이저
  - 운영체제 위에 존재
  - 다양한 변경이 가능하다
  - Hosted 하이퍼바이저

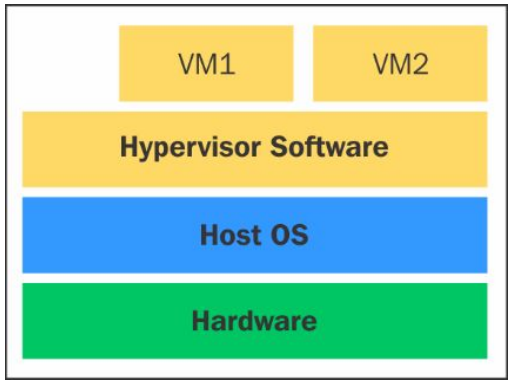
Type1 , Type2 하이퍼바이저

Type1



- 설치와 설정이 쉽다
- 사이즈가 작고 자원사용 최적화
- 부하가 작고 설치 애플리케이션이 작다
- 별도의 프로그램 및 드라이버 설치가 불가능

Type2



- 호스트 운영체제에 종속적이다.
- 광범위한 하드웨어 지원이 가능하다

가상화의 장점

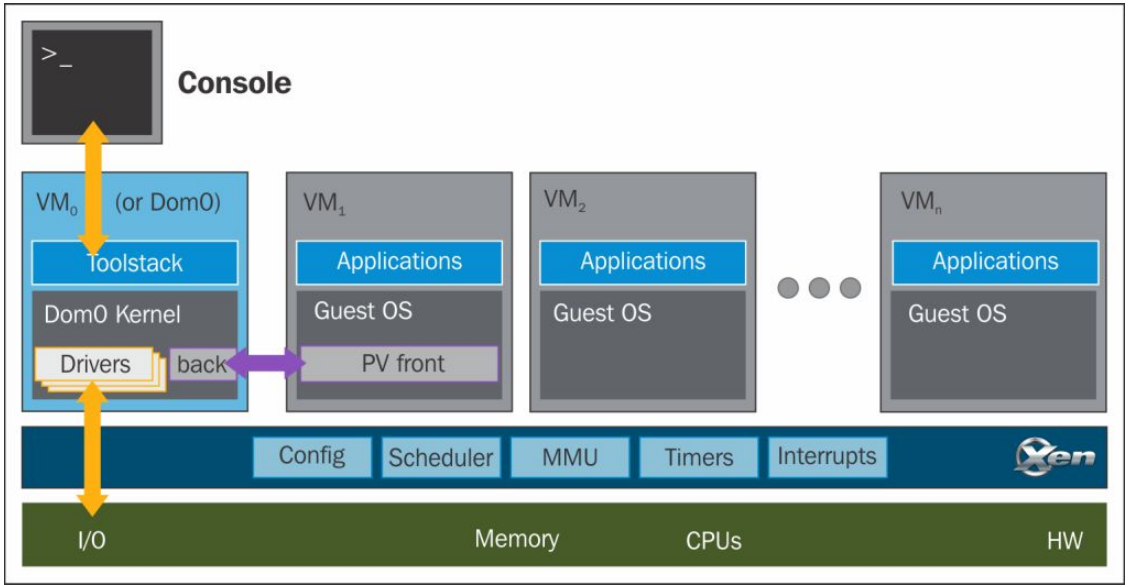
- **하드웨어 자원 효율성 향상**: 유휴 자원 최소화
- **비용 절감**: 서버 통합으로 인한 하드웨어, 공간, 전력 비용 감소
- **관리 용이성**: 중앙 집중식 관리 가능
- **유연성 및 확장성**: 빠른 시스템 배포 및 확장
- **격리성**: 장애 발생 시 다른 시스템에 영향 최소화
- **재해 복구**: 백업 및 복구 프로세스 단순화

오픈소스 가상화 프로젝트

Project	Virtualization Type	Project URL
KVM (Kernel-based Virtual Machine)	Full virtualization	http://www.linux-kvm.org/
VirtualBox	Full virtualization	https://www.virtualbox.org/
Xen	Full and paravirtualization	http://www.xenproject.org/
Lguest	Paravirtualization	http://lguest.ozlabs.org/
UML (User Mode Linux)		http://user-mode-linux.sourceforge.net/
Linux-VServer		http://www.linux-vserver.org/Welcome_to_Linux-VServer.org

Xen

- 케임브리지 대학 연구 프로젝트로 시작, 2003년 공개
- 2013년 4월 Linux Foundation 공동 프로젝트 로 이전



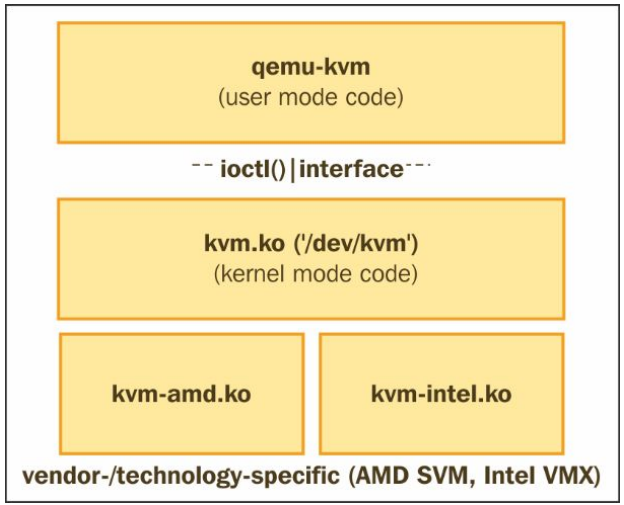
Xen

- **반가상화, 전가상화, 하드웨어지원 모드 사용 가능**
- **Guest VM 을 도메인 이라고 부른다.**
- **Xen 에는 두 종류의 도메인이 존재**
  - Dom 0 : 특권이 있는 도메인, 기능이 확장된 특수 Guest VM
  - Dom U : 권한이 없는 도메인, 일반 Guest VM
- **Dom 0**
  - VM을 만들고, 삭제하고, 관리 및 설정가능
  - 일반 Guest 시스템이 가상화 드라이버를 통해 하드웨어에 직접 접근 가능하도록 지원
  - API 인터페이스를 통한 시스템 관리기능 제공
  - 시스템에서 첫번째로 시작되는 도메인
  - Xen 프로젝트 하이퍼바이저를 위한 필수 도메인

KVM

- KVM (Kernel-based Virtual Machine)
- 하드웨어 가상화 지원기능 (VT-x, AMD-V) 를 활용한 최신 하이퍼바이저
- KVM은 KVM 커널모듈 설치로 하이퍼바이저로 변환가능
  - 리눅스 표준 커널에 KVM 커널 모듈을 추가
  - 표준 커널의 메모리 지원, 스케줄러 등을 사용가능한 장점이 있음
  - 리눅스 컴포넌트 최적화는 하이퍼바이저와 리눅스 Guest OS 둘다 이점을 갖음
- I/O 에뮬레이션을 위해 QEMU 를 사용
  - QEMU는 하드웨어 에뮬레이션을 지원하는 사용자 영역 프로그램
  - 프로세스, 디스크, 네트워크, VGA, PCI, USB, Serial/Parallel 포트 에뮬레이션

KVM 심층설명



- qemu-kvm 프로세스는 각 가상머신을 위해 별도로 실행됨
- virsh , virt-manager VM 관리도구는 libvirted를 통해 실행됨
- 가상머신 자원은 /etc/libvirtd/qemu 디렉토리에 xml 파일로 정의됨

KVM 심층설명

```
qemu 14644 9.8 6.8 6138068 1078400 ? Sl 03:14 97:29 /usr/bin/qemu-system-x86_64 -machine
accel=kvm -name guest1 -S -machinepc--m 5000 -realtime mlock=off -smp 4,sockets=4,cores=1,
threads=1-uuid 7a615914-ea0d-7dab-e709-0533c00b921f -no-user-config -nodefaults -chardev
socket,id=charmonitor-drive file=/dev/vms/hypervisor2,if=none,id=drive-virtio-disk0,
format=raw,cache=none,aio=native -device id=net0,mac=52:54:00:5d:be:06
```

- KVM 하이퍼바이저의 VM 프로세스
- -m 5000 : VM을 위한 메모리 5G 정의
- --smp=4,cores=1 : 각 소켓에 하나의 코어를 할당, 4개의 소켓

클라우드에서 리눅스 가상화가 제공하는것

리눅스는 클라우드 기반의 솔루션 개발을 위해 첫번째로 선택되어 왔다

- 아마존 EC2 클라우드는 Xen 가상화를 사용
- 디지털오션 KVM 사용

리눅스 가상화를 사용한 오픈소스 IaaS 클라우드 소프트웨어

- 오픈스택
  - Openstack Foundation 에 의해 관리되는 오픈소스 IaaS 클라우드 솔루션
  - 몇개의 오픈소스 프로젝트 컴포넌트로 구성
  - KVM 을 기본 하이퍼바이저로 사용
- 클라우드스택
  - Apache Software Foundation 에 의해 관리되는 IaaS 클라우드 솔루션
  - 아마존 EC2/S3 API 와 호환성
  - Xen 을 기본 하이퍼바이저로 사용

운영체제 가상화 / 파티셔닝

- 동일한 물리적 호스트가 서로다른 작업을 가능하게 함
  - 작업은 동일한 운영체제에서 독립적으로 작동
- 컨테이너 가상화
  - 물리적 서버에 다중의 격리된 운영체제 인스턴스(컨테이너)를 실행
- 컨테이너 가상화 종류
  - Solaris 컨테이너, FreeBSD jails, Parallels OpenVZ
- 단일 시스템에서 실행
  - 프로세스 격리 와 자원 관리는 커널이 담당
  - 컨테이너는 자신의 파일시스템, 프로세스, 메모리, 디바이스가 할당됨
- 멀티 OS 실행 제한
  - 윈도우, 리눅스, 유닉스등 다중 운영체제가 실행되는 가상화가 아님
  - 단일 OS 실행으로 성능과 효율성이 뛰어남

Type 2 하이퍼바이저

특징: 호스트 운영체제 위에서 실행

장점: 설치 및 사용 용이성, 개발/테스트 환경에 적합

사용 사례: 개인용 컴퓨터, 개발 환경

주요 제품:

- Oracle VirtualBox
- VMware Workstation/Fusion
- Parallels Desktop (Mac)
- QEMU

Type-2

- VirtualBox
  - 무료이며 코드가 공개되어 있다.
  - Oracle사에서 만듦
  - MAC/Linux/Windows모두지원
- VMWare player
  - 개인사용 무료
  - VMWare사 제품
- Pallels
  - MAC만 지원
  - M-series도 지원
  - Linux를 설치할수 있는 제품은 **퓨젼**이라함

컨테이너 가상화란?

정의: 애플리케이션과 그 종속성을 하나의 독립적인 단위로 패키징하는 기술

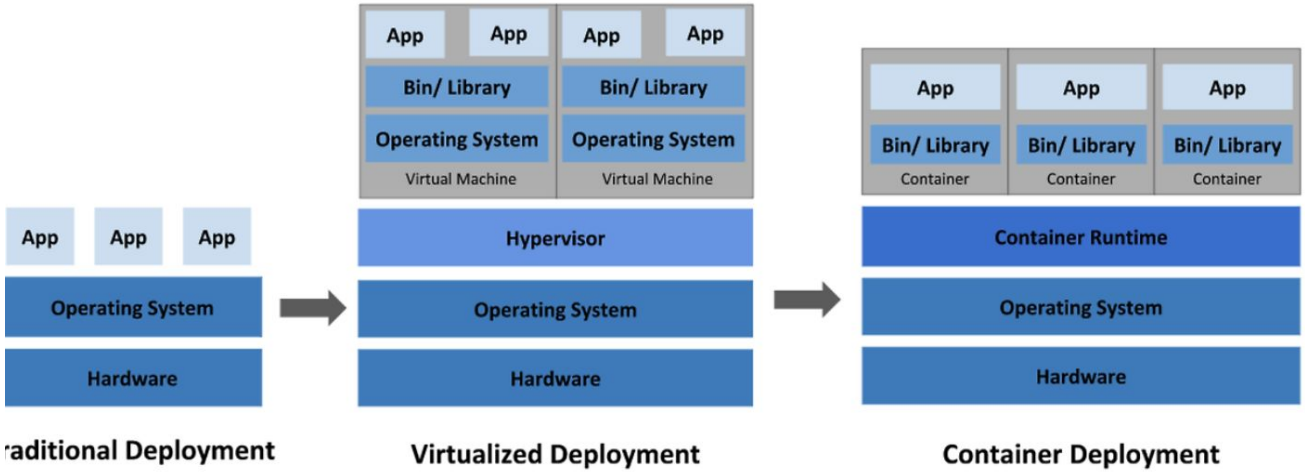
작동 원리: 호스트 OS 커널을 공유하면서 격리된 환경 제공

특징: 가볍고, 이식성이 높으며, 빠르게 시작 가능

특징

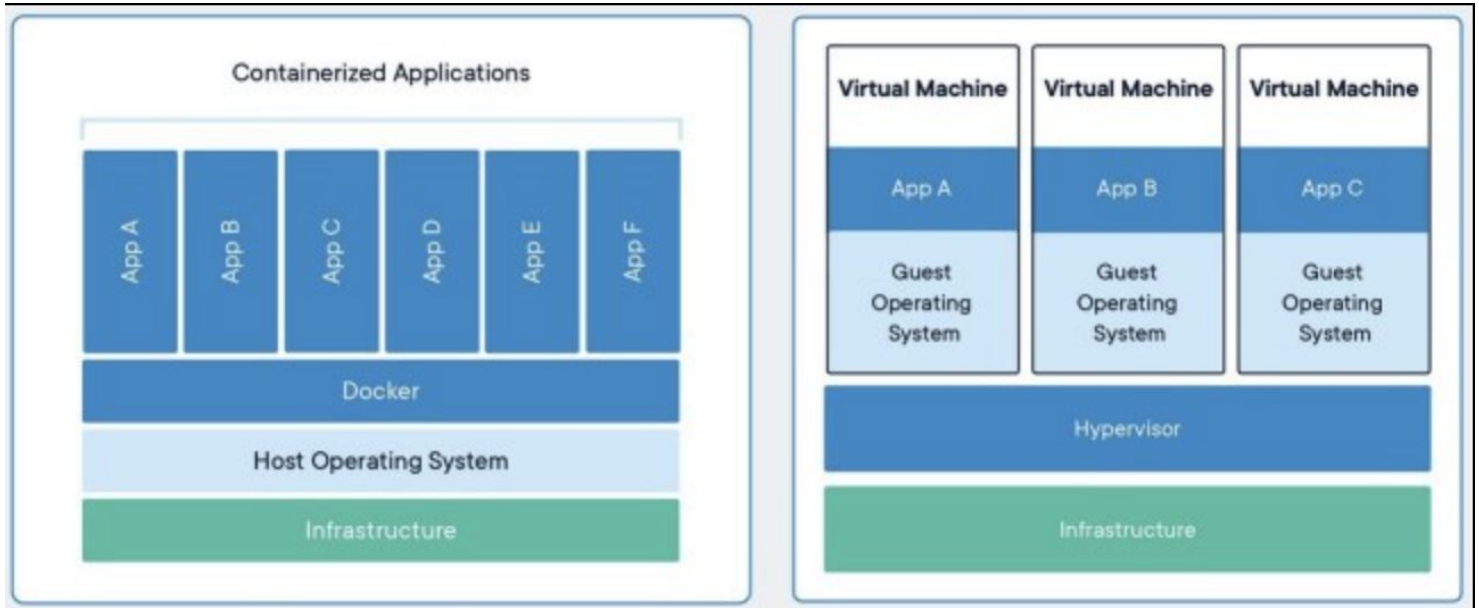
네임스페이스 (Namespaces): 프로세스, 네트워크, 파일 시스템 등 시스템 자원을 격리하여 각 컨테이너가 독립적인 환경을 가지도록 합니다.
cgroups (Control Groups): CPU, 메모리 등 시스템 자원의 사용량을 제한하고 관리하여 컨테이너 간 자원 경쟁을 방지합니다.
컨테이너 이미지 (Container Image): 애플리케이션 실행에 필요한 파일, 라이브러리, 설정 등을 포함하는 패키지 형태로, 컨테이너를 생성하는 데 사용됩니다.
컨테이너 런타임 (Container Runtime): 컨테이너 이미지를 실행하고 관리하는 소프트웨어입니다 (예: Docker, containerd).

컨테이너가상화


App
App
App
Operating System
Hardware
raditional Deployment
App
App
App
App
Bin/ Library
Bin/ Library
Operating System
Operating System
Virtual Machine
Virtual Machine
Hypervisor
Operating System
Hardware
Virtualized Deployment
App
App
App
Bin/ Library
Bin/ Library
Bin/ Library
Container
Container
Container
Container Runtime
Operating System
Hardware
Container Deployment

컨테이너가상화 (cont.)

- 운영체제위에 올리는 경우



끝