열한번째 주 첫째 날 가상화 복습 + 오픈스택 설치, 가상화와 오픈스택의 차이점

노트북:	필기노트		
만든 날짜:	2019-08-05 오후 10:11	**업데이트:**	2019-08-06 오후 9:17
작성자:	이종민		
태그:	오픈스택		

47day

오픈스택
amazon, ms, google 등 퍼블릭 클라우드 상품인 azure, amazon cloud 와 비슷하다.

네트워크, 라우팅, 스위치, 리눅스를 총 동원해야 내용을 이해할 수 있다.
오픈스택은 상당히 큰 프로젝트이다.

kvm에서 ovirt와 openstack의 차이점

Cloud가 오픈스택과 다른 점
On Demand (클라우드 특징으로 볼 수 없다) -> Self Service 필요할 때 만들어 쓰는 것

가상화는 가상컴퓨터를 만드는 건 관리자가 할 일, 일반 시스템 관리자가 하는 일인데 클라우드는 자기가 만들어서 자신이 쓰는 것이다.

클라우드의 모든 데이터 관점이 사용자 관점이다.
가상화를 올려서 인터페이스를 제공하는게 목적 = 똑같이 가상화이다.
결론은 가상화라는 기술을 쓰는 건 똑같은데, 직접 구성하느냐 받아서 쓰느냐의 차이같다.

클라우드 구현하는 업체, 고객의 입장, 퍼블릭 클라우드 사용 (아마존, 애저 등) or 프라이빗 클라우드 사용 (회사 내부에서 사용)

관점은 사용자 입장밖에 나오지 않는다.
사용하는 목적, 방법 자체가 틀리다. 우리가 직접 구현해서 서비스를 하느냐, public, private 으로 self service 구성하느냐 차이

가상화와 클라우드 구별은 인프라의 레벨에서 구분한다.
시스템 동작하기 위해서는 하드웨어가 필요한데, 하드웨어를 추상화(가상화) 한다.
하드웨어를 가상으로 만들었기 때문에 인프라를 가상으로 제공해준다. (어느적으로 물리적으로 여유는 있어야한다)

아마존, 구글, ms, 네이버 등은 인프라를 제공해주는 서비스이다.
중요한건 가상의 자원을 어떻게 쓰느냐이다. 클라우드는 셀프 서비스이기 때문이다.
누가 어떻게 쓰느냐에 따라 방법 자체가 틀리다.

가상화
가상화 나오기 이전, 시스템이 나온 처음 시점부터 가상화는 존재했었다.

isolation - 격리
하드웨어 위에 커널 -> 프로세스 끼리 격리가 안되서 서로 영향을 끼치고 커널, 시스템까지도 영향을 미칠 수 있다.
크롭션 (충돌) 자기한테만 문제가 있어서 종료가 된다면 상관없지만, 다른 프로세스에 영향을 미치게 되면?

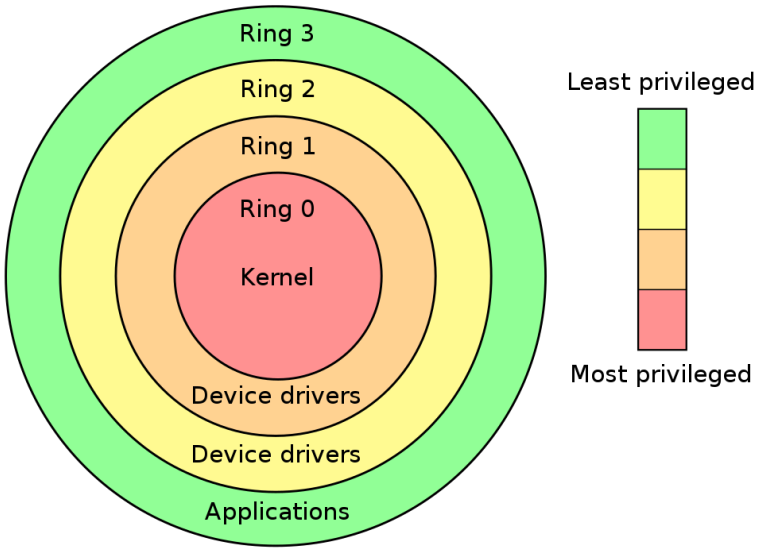
프로세스 취약점을 이용해서 다른 프로세스로 침입 한다던지. 충돌로 인해서 격리가 안되서 다 죽을 수 있다.

네트워크와 아무 상관없이 내 컴퓨터 고장이나도 다른 컴퓨터에 영향을 주지 않은거처럼 분리가 되면 영향이 없다.
그래서 컴퓨터 한 대의 하나의 시스템만 올려서 사용했었다. 그렇게 사용하면 컴퓨터 자원이 낭비가 되어서 가상화를 하게 되었다.
소프트웨어 격리, 운영체제가 다르면 된다. << -- $ 이부분 피드백이 필요 뭐가 다르면 되는지. $
하드웨어 위에 커널을 여러 개 올려서 여러 개의 프로세스를 작동 시키는 것 = vm 하드웨어를 분리 시켜야한다.
분리를 시키는 기법이 2가지가 있는데, 하드웨어 레벨에서 분리시키는 방법과 소프트웨어 레벨에서 분리 시키는 방법이 있다.

hw - 하드웨어 파티션
펌웨어 방식 (소프트웨어), 하드웨어에서 하드웨어를 나누는 방식 // 펌웨어 : 하드웨어를 제어하는 가장 기본적인 소프트웨어
하드웨어를 직접 나누는게 가장 좋은 방법이다. (대용량, 유닉스 방식은 다 하드웨어 파티션을 사용한다) / 하드웨어 자체에서 나눌 수 있는 기능이 있다.
우리가 사용하는 x86은 하드웨어를 나누는게 불가능하다. (목적이 다르기 때문에)
인텔에서는 하드웨어 파티션을 못하는 이유가 cpu 아키텍쳐를 바꾸면 운영체제도 바꿔야하는데, 인텔은 운영체제가 없어서 못한다.

sw - 소프트웨어 파티션, 처음에는 하드웨어 위에 하이퍼 바이저 형식으로 했었다.
하이퍼바이저는 실제 하드웨어에게 다시 명령을 내려줘야하는데, 어플리케이션, 커널, 하이퍼바이저, 하드웨어 구조로 있었는데링 구조가 다른데 하이퍼 바이저를 배치를 시킬까? -> 처음에는 ring1에 하이퍼바이저가 있었다. zero는 커널이 될 수 밖에 없기 때문이다.
커널은 여러가지 커널을 올릴 수 있다. 하이퍼바이저는 binary translation 명령 변환을 시켜줘야한다. (커널이 준 명령 변경)
bianry transation = emulation 1:1로 대응이 된느 다른 명령어로 바꿔줘야한다. 에뮬레이터와 개념이 유사하다.
가상화의 기본적인 개념은 바이널리 트렌젝션, 명령어를 계속 변환 시켜주는 것이다. 격리가 안된다.

hypervisor -> os
운영체제 입장에선 위로 어플리케이션 아래로는 하드웨어가 있어야한다. application, hw를 제어하는게 운영체제 이다.
하드웨어인거처럼 application을 속인다. / 추상화, 자기가 가상으로 사기친다.



protection ring - 보호링

리눅스에서 네트워크가 안잡혀서 드라이버를 설치한 적이 있다.
r8168-dkms 네트워크 드라이브를 설치 하기 위한 소프트웨어, 커널을 통해서 하드웨어에 접근한다.

hw -> os -> app
운영체제 설치 시 위에는 어플리케이션 아래에는 하드웨어가 있어야한다.
하드웨어가 그렇게 설계가 되어있고, 거기에 맞게 운영체제가 설계가 되어있으므로 (보호링 그림 참고)

full virtualaztion - 전체를 다 가상화 한다. binary transltation (sw) - 성능이 별로 좋지 않다. / vmwere, ms
할 일이 많다보니 병목현상이 일어나서 성능 저하가 있다. 윈도우를 올릴 수 있는 특징이 있다.
hardware assisted (hum) - 하드웨어의 도움을 받아서 translation을 한다. < 현재의 전가상화 형태
intel vt-x
amd amd-v

para virtualization - 일부분만 , 필요한 것만을 가상화 한다. xensource
커널이 hypervisor를 알아야한다. = 커널의 기능이 바뀌어야한다. = 커널 수정이 필요하게 되는데 리눅스와 일부 유닉스만 가능하다.
윈도우는 커널을 공개하지 않았기 때문에 불가능하다 = 커널을 수정할 수 없기 때문에 불가능.
전가능화보다는 성능이 좋다.
현재는 para 보다 full을 사용하고, 이론 상 para와 차이가 미미하다.

kvm
초기에는 성능이 좋지 못했다. xen이 인수 당한 후 리눅스는 kvm 밖에 남지 않았고 그래서 kvm이 성장하게 되었다.
QEMU
KVM
kvm은 2가지로 나뉜다. 원래 2개가 별개 였었다. para, full 기능을 갖고 있었다.
xen을 빼고 가상화 회사는 vmware, ms, kvm 밖에 남지 않았다.

vm은 cloud 기술이 없었으므로 지금은 많이 힘들었다.
현재는 para 방식은 xen 에서만 사용하는 기술이 되었다.

grep vmx /proc/cpuinfo
intel cpu가 지원하는 기술을 볼 수 있는 명령어

grep svx /proc/cpuinfo
amd cpu가 지원하는 기술

lscpu
cpu 보는 명령어

vmware esxi -> vcenter
micro soft hyper-v -> scvmm
linux kvm, xen -> ovirt, xencenter

type1 (native, baremetal) hw -> h.v -> vm
하드웨어 위에 하이퍼바이저 그 위에 vm이 있는 형태
하드웨어 깡통 위에 하이퍼 바이저가 올라가는 형태이다.

원격에서 여러 시스템을 등록하고 로컬 하이퍼바이저를 관리하는 형태이다.
로컬 하이퍼바이저로 관리하는 형태라서 로컬에 저장이 되어있다.
기본적으로 무료이다. 단 자기가 자신의 시스템을 사용한다 = 자기껏만 관리가 가능한 형태
개별적으로 관리를 하게 되면 한도 끝도 없고, 옮기는 것도 쉽지 않다. => cluster 기능을 지원하지만, 판매상품이라고 한다.

물리적인 시스템을 논리적으로 관리하는 방식이다.

cluster는 두 가지 방식이 있는데
HPC (high perpomance computing) - 고 성능의 컴퓨터에 사용, 병렬 컴퓨팅 // 시뮬레이션, 슈퍼 컴퓨팅 형태
HA (high avilabliity) - 고 가용성
통합 네트워크를 지원, 물리적으로 떨어져 있지만 논리적으로는 같은 네트워크에 있는 거 처럼 사용 // 우리가 kvm으로 하는 형태와 유사하다.
tunneling

마이그레이션은 두 가지가 있다
hot migration(live migration) - stoage를 구성, vm의 정보를 storage에 저장. 소유권 이전만하면 vm을 이전 시킬 수 있다. // memory 동기화를 시켜야한다.
sql, in memory db 메모리에서만 변경, 마이그레이션 이전
cold migration - 시스템이 종료된 상태에서 데이터를 이전 시키고 소유권을 이동한다.

vmware workstation (player)
funsion
orcle virtualbox

type2 (hosted) hw -> os -> h.v
하드웨어 위에 일반 os (host os)가 있는 형태
host os 위에 올라가는 하이퍼 바이저 형태라서 hosted 이다.
단계가 많아진다. = 성능이 안 좋다라는걸 의미한다.
테스트, 개발 용도로만 사용한다.

cloud는 on demand 내가 필요한 만큼 직접 구성해서 사용하는 형태
벤더 입장과 사용자 입장이 다르다.

벤더가 어디까지 구현을 해주는가?

HW -> OS -> A
벤더에서 인프라만 구현해서 제공을 해주나? -> 사용자는 제공해준 인프라를 가지고 os를 설치한다

on-premises = 직접 소유하고 있는 형태
iaas : 벤더가 인프라를 제공해준다. (플랫폼을 개발, os 설치 등)
paas : 벤더가 개발 플랫폼까지 제공, 고객은 소프트웨어를 개발해서 사용 or 다른 고객에게 제공
saas = 벤더가 소프트웨어까지 제공해주는 형태

아마존, 구글, ms은 기본적으로 인프라를 제공해주는 것 + 여러가지 개발환경, 소프트웨어까지 제공해준다.
오픈스택의 가장 기본, 클라우드 서비스의 기본은 인프라 시버스가 기본이다.

saas = google docs를 예로 들 수 있다. 소프트웨어를 제공

하이퍼바이저랑 다른건 self service 밖에 없다. 다른 비교는 크게 의미가 없다.

openstack= private cloud service -> 말 그대로 직접 구성하고 회사 내에서만 사용하기 때문에..
openstack은 계정을 만들 수 있는 기능이 없다. 사용자가 직접 만들 수 없다. 관리자가 만들어줘야 함
그리고 과금 기능이 없다. 사용자마자 사용량을 체크해서 과금해야하는데, 이런 부분을 개발해서 판매? 하는거 같다.

요즘은 multi cloud 특정 회사 제품 하나만을 사용하지 않는다. server가 down을 할 수 있기 때문에..
얼마전에도 아마존 리전이 다운 됐었다고 한다.
특정 회사만을 사용하는 경우에 대규모 장애에 대비하기 힘들다.

기능을 구분해서 service를 올리거나, 똑같은 서비스를 여러 개의 cloud에 올린다거나 아니면 public + private 형태로 많이 사용한다.