

다섯 째 주 넷째 날 부팅 프로세스, 소프트웨어패키지, 네트워크 관리

노트북: 필기노트

만든 날짜: 2019-06-27 오후 11:31

업데이트: 2019-06-30 오후 12:28

작성자: 이종민

태그: 리눅스

20DAY

예전 리눅스 1번 포르세스가 init으로 되어있었다. / 지금은 systemd 이다.

BIOS : 하드웨어 관련된 설정을 제어 할 수 있다.

메인보드에 바이오스에 관한 칩셋이 있다.

BIOS는 하드웨어 기능을 제어, POST, 부팅 장치의 순서를 지정 할 수 있다.

BIOS는 MBR 방식만을 지원한다.

CMOS : 바이오스가 칩으로 있는데 그 칩에 대한 공정을 CMOS라고한다. / 옛날 카메라의 이미지 센서도 CMOS 공정으로 만들어졌다.

CMOS칩에 저장 되어있다.

UEFI : 바이오스의 차세대 기능 / 목적은 같다. POST를 하고 부팅 장치의 순서를 결정한다.

그러면 왜 만들었을까? GPT지원이 안되서 GPT 지원하려고 만들었다.

Flash memory에서 소프트웨어를 읽어서 실행한다. (하는 역할이 post, 부팅 장치 결정)

SDA에서 bootloader를 가동시킨다.

윈도우는 ntloader, 리눅스와 유닉스는 GRUB라는 bootloader가 있다.

부팅장치에서 부트로더를 읽는다. 부트로더 같은 경우 커널을 실행한다.

파티션을 나눠놓은 이유가 옛날 file system은 잘 깨졌는데, 파티션을 나눠놓지 않으면 파일시스템이 깨지면 시스템 전체가 깨졌기 때문에

부트로더의 파일시스템이 깨지더라도 시스템에 영향을 미치지 않게 따로 나눠놓았다.

현재는 file system이 깨지는 일은 거의 없다고 한다.

cd /boot/ - boot dirctory

현재 리눅스는 grub2를 사용한다.

디렉토리 안에 커널이 존재한다. (initramfs 라는 이름으로 저장 되어있다. root file system)

cd /boot/grub2

menuentry 처음 부팅 시 grub에 보이는 3개의 메뉴가 있는데,

시스템을 처음 설치하면 메뉴는 2개 밖에 안나온다.

그럼 왜 3개가 있을까?

처음에 드라이버를 설치하면서 커널 패키지를 설치했는데 새로운 커널을 설치 했기 때문에 자동으로 메뉴가 늘어난거다.

시스템을 잘 다루면 수정 할 수 있지만, 내용도 방대하고 어렵다.

linux 16 - 커널파일 / 커널을 어떻게 실행할건지에 대한 옵션 (모르고 건들면 안된다)

linitrd 16 - 루트 파일 시스템 / 커널을 어떻게 루트에 올리건지에 대한 옵션

resuce - 윈도우의 안전모드랑 비슷하다고 보면 된다.

cd /etc/default

grub file이 있는데, timeout=5 가 설정되었다. 처음 부팅 시 5초안에 반응 안하면 자동으로 전원이 켜지게 설정 해놓은것

/etcgrub.cfg

파일 처음에 do not edit this file - 함부로 수정하지말라고 적혀있다.

grub2-mkconfig에 의해 설정파일이 만들어진다. -> /boot/grub2/

파일들을 문법체크를 해서 읽어준다.

우리가 grub2를 직접 제어할 일은 없다.

커널이 설치가 되면 해당되는 새로 설치된 커널을 실행하기 위해서 grub2에 메뉴를 만들어준다.

Runlevel은 아래부터 차근차근 올라간다.

0 - 시스템 종료

1 - 복구 셸 이라고 하는데 single-user-mode 이다. / 파일 시스템을 마운트하고 관리자 말고 접근도 불가능하다. / 네트워크 기능이 없다.

2 - 멀티 유저 모드인데, 네트워크 기능이 없다.

3 - 멀티 유저 모드에 네트워크 기능이 있다.

4 - 안 쓴다

2,4번은 사용하지 않는다. 일반적으로 3번 사용

1번이 실행되고 기본적으로 3번형태로 차근차근 올라온다.

5 - gui를 서비스 해준다.

6 - 재부팅, 0번으로 갔다가 3번으로 가는 형태를 6번이라고 할 수 있다.

local-fs.target :

/etc/fstab에 등록된 마운트 정보로 파일시스템을 마운트한다. / 디스크 마운트를 시킨다.

root를 제외하고 마운트한다. / 이미 swap과 root는 마운트가 되어있다.

sysinit.target :

시스템 마운트, 스왑, 커널의 추가 옵션 등 여러가지 기본서비스를 실행한다.

basic.target :

firewalld, microcode, SELinux 등 방화벽 커널 메시지와 시스템 중요 서비스를 실행하거나 모듈을 로드한다.

default.target :

multi-usertarget, graphical.target 등 해당되는 서비스를 실행한다.

중간에 문제가 발생하면 로그인 프롬프트가 뜬다. (네트워크가 안된다)

보통 파일시스템, 마운트 문제, 마운트 지정을 잘못하거나 없는 경우 로그온 프롬프트가 뜬다.

graphical.target :

gui를 지원하는 다중 사용자 모드이다. init 프로세스의 런 레벨 5레벨과 매핑되며, runlevel5.target으로 사용할 수 있다.

multi-user.target :

커맨드 라인을 제공하는 다중 사용자 모드이며 init 프로세스의 런 레벨 3과 매핑되며,

runlevel3.target으로 사용할 수 있다.

systemd 부팅 절차

BIOS/UEFI -> BootLoader -> Kernel -> systemd -> default.target -

graphical.target

local-fs.target -> sysinit.target -> basic.target -> multi-

user.targ

systemctl poweroff | systemd에서 시스템 종료 할 때 사용하는 명령어
systemctl reboot | systemd에서 시스템 재부팅 할 때 사용하는 명령어

우리가 자주 사용하는 poweroff, reboot은 systemctl 명령어에 심볼릭 링크가 걸려있다.

halt.target :

운영체제는 종료되는데 전원은 켜져 있는 상태이다. 일반적으로 잘 사용하지 않고, 서버에서 쓸 경우가 있기도한다.

kexec.target :

인터넷을 찾아보면 재부팅을 빠르게 할 수 있습니다라고 하는데, 커널만 재실행 커널을 교체했을 때 업데이트하기 위해 재부팅 할 때 사용한다.

하드웨어를 추가 했을 때 kexec를 하면 안된다. 재부팅을 해줘야한다.

systemd 타겟 유닛, 런 레벨

- 0 poweroff.target 시스템 셧다운하고 종료한다.
- 1 rescue.target 복구 셸(rescue shell)
- 2 multi-user.target 커맨드 환경의 다중사용자 모드
- 3 multi-user.target 커맨드 환경의 다중사용자 모드
- 4 multi-user.target 커맨드 환경의 다중사용자 모드
- 5 graphical.target 그래픽 환경의 다중사용자 모드
- 6 reboot.target 시스템을 셧다운하고 재부팅

3번하고 5번을 가장 많이 사용한다.

emergency.target :

가능한 최소한의 환경을 제공하는 긴급 셸, 접근하려면 root passwd가 필요로함.

특수타겟 장애처리를한다. (t.s 트러블 슈팅) | rootfile system이 read only 상태이다.

파일의 내용 수정이 필요하다면 루트 파일시스템을 rw로 마운트를 해야 가능하다.

rescue.target :

단일 사용자 환경을 제공하는 복구 셸, 접근하려면 root passwd가 필요로하다.

emergency와는 다르게 rw 상태이며, 네트워크는 비활성화 되어있다.

sysinit.target과 rescue.service가 활성화 되어야 rescue가 활성화된다.

multi-user.target :

일반적인 CLI 환경

graphical.target :

일반적인 GUI 환경

who -r = runlevel / 사용 중인 런 레벨 타겟 확인, 볼 일도 없고 실제로 안쓰는 명령어

systemctl set-default [multi-user.target , graphical.target] 등으로 명령어를 입력해서 시스템 부팅될 때 CLI, GUI 부팅 설정을 바꿀 수 있다.

systemctl isolate [multi-user.target , graphical.target] level 전환 / 사전적의미로 격리하더라는 의미를 갖고있다.

시스템 부팅 중 타겟 지정도 가능하고 e키를 눌러 편집(edit)모드도 가능하다.

루트 패스워드 복구

커널 항목에서 linux16으로 시작 하는 라인의 맨 마지막에 램 디스크 초기화를 중단하는 옵션인

rd.break를 입력한 후

switch_root 셸로 진입해서

switch_root# mount | grep -w 'sysroot' // 지금 switch_root는 실제 루트 파일 시스템이 아니다.

switch_root# mount -o remount rw /.sysroot // 실제 사용하는 루트를 rw로 다시 마운트로 한 후

switch_root# chroot /sysroot // 실제 root 파일과 switch_root를 교체한다.

passwd // 명령어 입력 후 패스워드를 설정하면 root 패스워드를 바꿀 수 있다.

touch /.autorelabel / SELinux 재생성을 하면 적용이 된다. 없으면 안되고 원래대로 돌아간다.

파일시스템 문제 복구

복구 셸로 접근해서 /etc/fstab 파일을 수정한다.

소프트웨어 패키지

옛날 리눅스는 소프트웨어를 설치하기 위하여 아카이브 파일, 압축파일로 풀어서 컴파일해서 설치를 했다. / tar 아카이브 개념으로 설치, 제거

귀찮고 까다로워서 소프트웨어 패키지가 나왔는데 unix마다 패키지 구조가 다르다. / 레드햇 rpm, 우분투 deb

운영체제도 여러개의 패키지를 묶어 놓은 것을 설치하는 방식이다.

레드햇은 과거에 rpm 패키지를 사용했지만 여러 단점들에 의해 현재는 yum을 사용한다.

RPM 패키지 관리

httpd - 2.4.6 - 40.e17.centos .x86_64 .rpm

글자부터 글자로 끝나는 부분이 패키지의 이름이다.

대쉬 뒤 숫자 세 자리는 패키지의 버전을 나타내고

대쉬 뒤 글자, 숫자가 릴리즈 정보 / 리눅스의 버전이 명시 되어있다.

.(온점) 아키텍처 정보 / x86_64 x32 i386, 586 (32bit) noarch (아키텍처 구분하지 않는다.)

intel | amd i86_64 | i386, 586

ppc (ibm power cpu) arm (arm cpu / mobile)

.rpm : rpm의 확장자

rpm 쿼리 옵션

-a : 모든 패키지 확인

-f : 해당 파일이나 디렉토리가 포함된 패키지 확인

-c : 설정 파일 확인

-d : 문서 파일 확인

-i : 자세한 정보 확인

-l : 파일 목록 확인

-s : 파일 상태 확인

-R : 종속성 패키지 확인

RPM

-O 파일명 지정하면 받아진다

-Uvh 설치 / lib가 있는 패키지를 찾아서 설치해야한다. (의존성)

의존성 때문에 귀찮아서 yum이 나왔다. yum은 인터넷에서 자동으로 찾아서 install 해준다.

리눅스마다 필요 패키지 개수가 다르다.

yum 리포지토리

yum repolist

base centos-7 - base 10019

extras centos-7 - extras 13237

updates centos-7 - updates 2137

mirrorlist : mirror server 갯아 가까운 위치에서 다운로드

baseurl : 주소를 직접 지정해놓은것

pgp : 서명

gpg (gnupg) : pgp를 사용할 수 있게 해주는 프로그램 / pgp 서명 확인을 자동으로 한다.
gpgkey : 서명을 검증하기 위한 공개키 / url 형식이다. 파일을 지정할 때 url 형식으로 지정할 수 있다.

enabled : linux source code

yum repolist all (명령어)

주소를 알고 있다면 repolist를 추가해서 설치할 수 있다.

우분투는 deb를 이용한다. deb 명령어 rpm 처럼 필요한 파일을 다 받아서 했는데 레드햇의 yum과 같은 apt라는 커맨드를 사용한다.

lock 걸리다. 터미널 여러 군대에서 동시에 사용하면 안된다.
packgekit

yum [subcommand] [argument = 패키지 이름]
설치 할 패키지가 repolist에 있는지 확인해야한다.

yum info [패키지명]

anaconda : 설치할 때 맨 처음에 나오는 메뉴, 도구를 anaconda라고 한다.

yum provieds / rpm -qF와 같다. : 관련된 패키지 확인
mirror : 명령어를 할 때마다 미러서버가 달라진다.

yum search : 검색, 패턴 | 잘 안써진다. 너무 많은 내용이 나온다. 패키지 목록이라던지 살펴볼 수 있다.

yum list [패키지명] :
search보다 선호해서 사용한다. 완전한 패키지 명이 들어가야한다.
아무것도 안나오면 repolist에 없는 내용이다. = rpm -와 같이 설치된 패키지의 목록이 나온다.

yum update [패키지명] : 설치된 모든 패키지 중 업데이트가 가능한 패키지를 업데이트 하는 명령어 / 설치가 안된 패키지면 설치 해준다.

yum [-y] remove [패키지명] : 설치 할 땐 의존성 관계에 있는 패키지를 다 같이 설치하지만, 삭제할 땐 지정한 패키지만 삭제된다.

yum group : 서브 커맨드에 서브커맨드 사용이 가능하다. 붙여서 써도 사용이 가능하다.
info
install

tail yum.log
yum history
rollback 4 = undo 5
redo : undo 했던 것을 다시 설치

httpd, mariadb, mariadb-server php vsftp bind 등 (우리가 사용할 패키지들)

/etc/yum.repos.d :
centos 패키지 목록

네트워크 관리
dns를 세팅 해야한다.
현재 virtualbox라는 가상화 프로그램을 쓰고 있고 가상의 네트워크를 이요하고 있다. /
virtual network
가상화 라고 하는 여러 프로그램들은 가상 네트워크를 구성할 수 있는데 프로그램마다 다 틀리다.
가상화 프로그램에서 네트워크 구조가 다 다르다.

virtualbox network 구성

버추얼 박스는 vm에 네트워크를 4개까지 꽂을 수 있다.

attached to : 어디에 연결을 할 것인가?

NAT :

NAT Network :

Bridge :

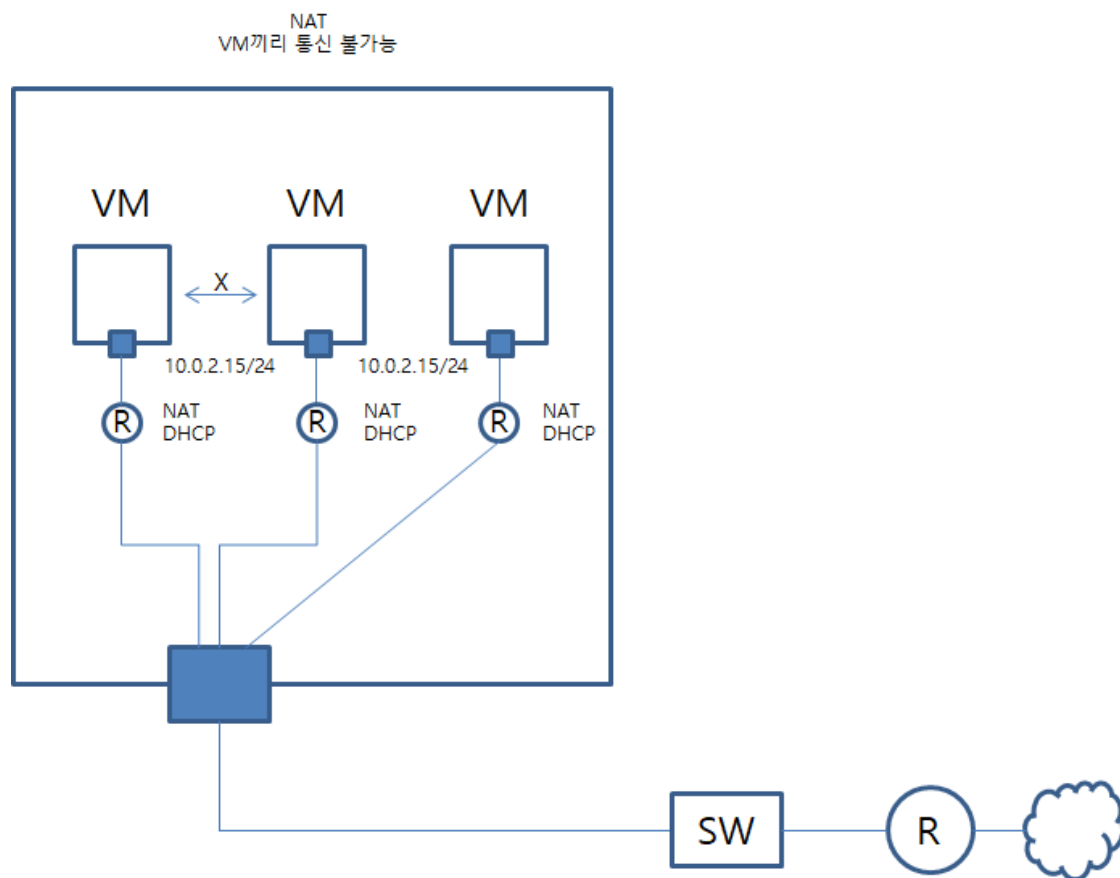
Internal :

Host-Only :

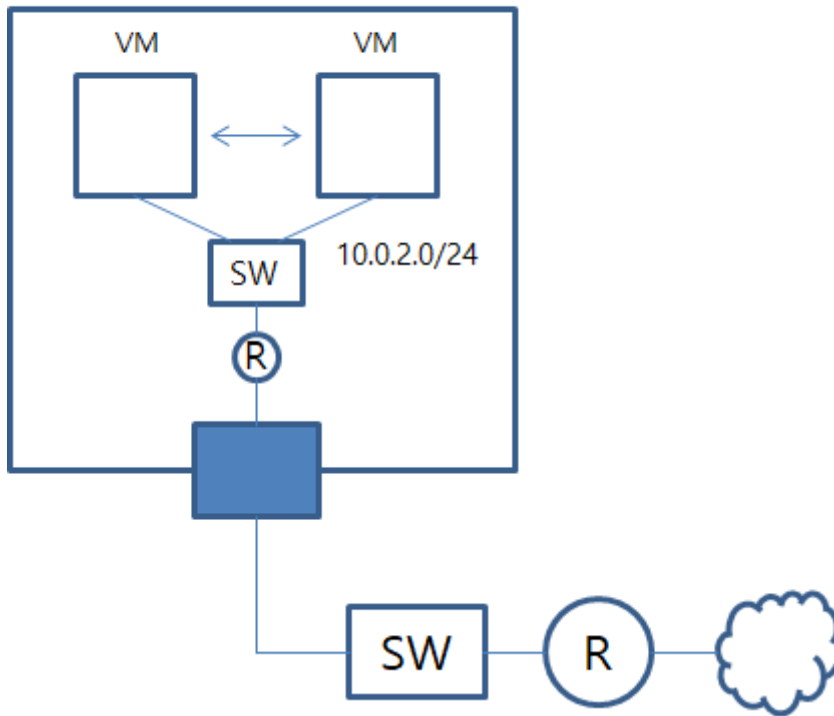
virtualbox -> tools -> network -> create -> host-only network

virtualbox file -> preferences -> create -> network -> nat network

cent os

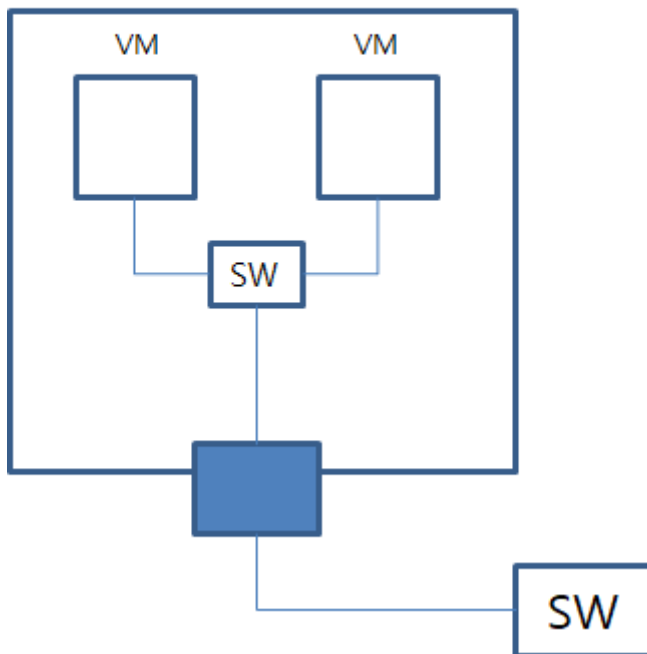


NAT Network
VM끼리 통신 가능



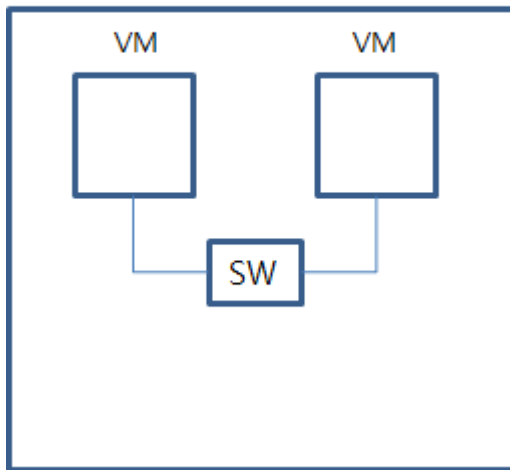
Bridge

네트워크 스택을 경유하여 NIC 통해 연결
IP 할당 제약이 없을 때 적합한 방식이다.

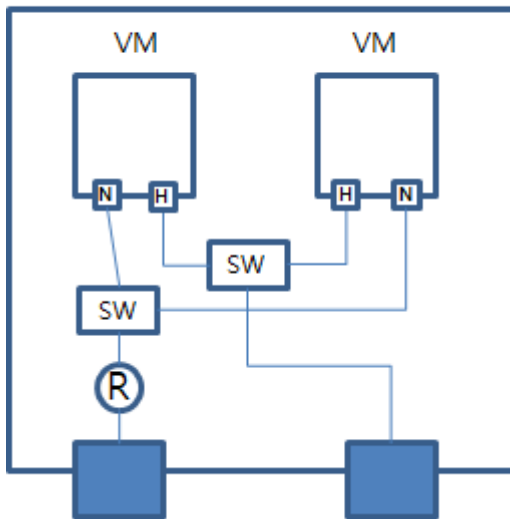


브리지 방식을 이용하면 실제 공인 IP를 여러개 부여 하여 서비스 할 수 있다.
VM이지만 실제와 똑같이 동작하는데, VM을 많이 띄우게 되면 IP가 부족해지고, 대역폭이 꽉차며
보안이 문제가 된다.
D-DOS 공격 받기 좋은 방식이다.

Internal
Vm끼리만 통신가능



Host only



Internal 방식

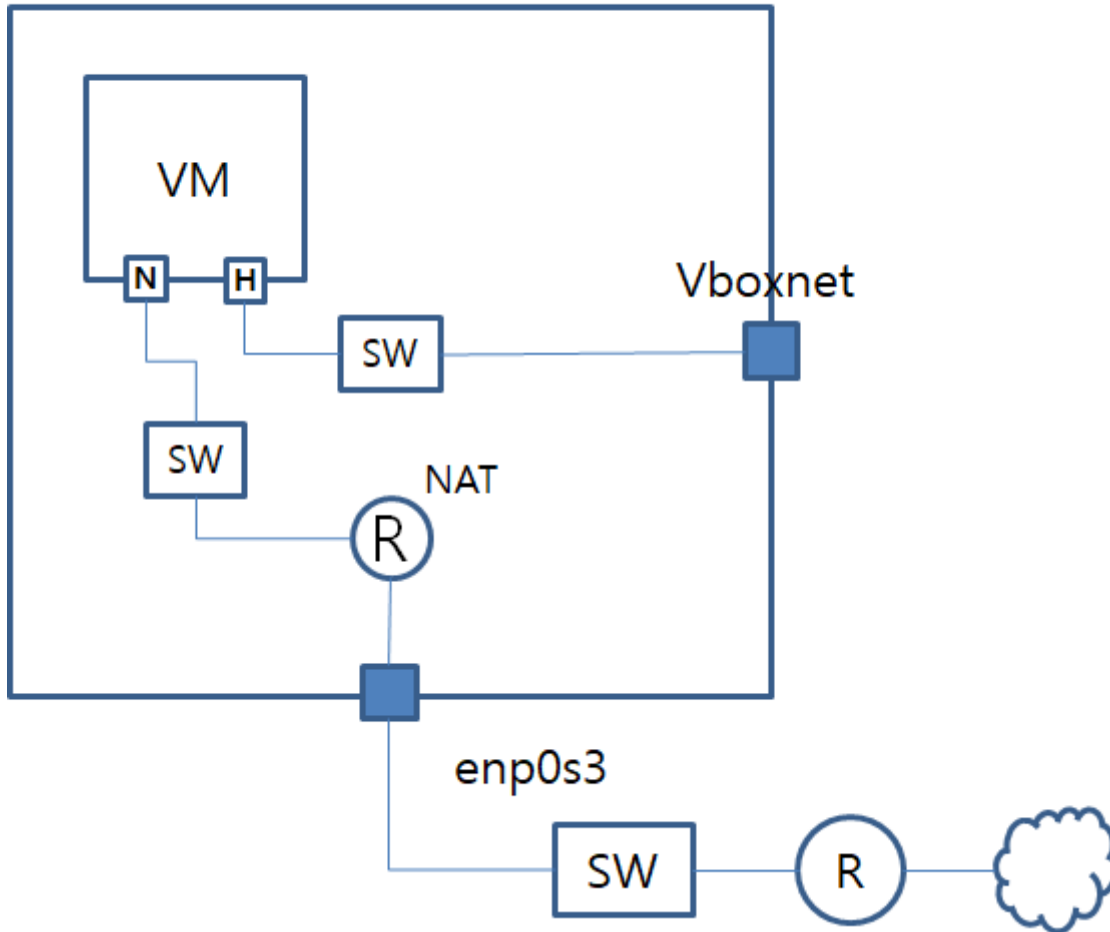
내부에 다수의 가상 시스템간의 통신을 지원하고 외부로부터 격리함으로써 보안을 강화할 수 있다.

호스트 네트워크와 분리되어 동작하므로 인터넷 연결이 안된다.

Host only 방식

내부 네트워크와 비슷하지만 호스트와 연결이 지원된다는 점이 다르다. 호스트와의 통신이 지원되고 dhcp를 이용한 ip 할당이 가능하다.

VM 구성 그림



vm의 구성 방식

옛날 리눅스는 인터페이스 이름으로 마운트 되어 있어서 장치를 다른 포트에 꽂으면 나중에 꼬이는 현상이 발생 했는데,
최신 리눅스는 디바이스 이름을 사용해서 연결을 하기 때문에 인터페이스 이름이 별로 중요하지 않다.

ifconfig :

옛날 방식의 네트워크 인터페이스 구성, 확인하는 명령어이다.

요즘 리눅스는 ifconfig가 기본으로 포함되지 않은 리눅스가 많다. nettools 사용 안하는걸 권장 / 제대로 된 정보를 표시를 못한다.

ip route :

라우팅 테이블을 확인하는 명령어

traceroute :

목적지까지 가는 라우터 경로를 출력하는 명령어

ping :

ICMP 패킷을 주고 받아 호스트 간 네트워크 연결을 확인하는 명령어

네트워크 관리자

nmcli 커맨드라인 도구

nmtui 테그슬 유저 인터페이스 도구

nm-connection-editor 그래픽 도구 (안 쓴다)

/etc/resolv.conf / G.W 주소, 가상 서버는 G.W가 dns가 된다.

인터페이스 이름

장치 유형 :

en(이더넷), wl(wlan), ww(wwan)

어댑터 유형 :

o(on-board), s(hot-plug-slot), p(pci위치), b(bcma bus core), ccw(ccw bus group)

변하지 않는다. 장치의 이름이 바뀌지 않는다.

레거시 네트워크 방식

networkmanager 를 stop, disable 하고 mask 한 후 ifcfg를 직접 수정하는 방식이다. / ip 명령을 사용하여 인터페이스 ip 주소 추가 등

ifcfg-*

ifcfg-enp0s3 ifcfg-lo

IP setting 하는 file vi 수정 하면 static, dhcp 가능하다.

수정을 하면 systemctl에서 network 를 다시 시작해주면된다. / 이 서비스는 중요한 서비스라서 꺼지면 안된다.

network 하는 서비스가 아래 파일을 참조한다.

/etc/sysconfig/network-scripts/ifcfg-*

nmcli

커맨드를 사용해서 네트워크를 만들수 있다.

보통 method, dns, addresses, gateway 등을 기본적으로 설정한다.

nmtui

nmcli 보다 간단하게 텍스트 상자 형식으로 ip를 설정할 수 있다.

hostnamecli set-hostname [원하는 호스트네임]

으로 호스트 네임을 설정할 수 있다. / nmtui를 사용하면 더욱 간단하게 설정이 가능하다.