2020/08/24~2020/12/24

융복합 프로젝트형 클라우드(MSA) 서비스 개발

오픈이지 박창렴 / myanjini@gmail.com / 010-2982-7033

좌석배치

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 유찬영 |  |  |  | 박진우 | 황동윤 |
| 김채린 | 김민지 |  |  | 강륜화 | 김유진 |
| 이민용 | 홍유진 | 방진실 | 김윤영 | 이동재 | 정유라 |
| 이준의 | 주성우 | 신호연 | 여찬진 | 이윤식 | 윤서율 |
| 안예진 | 배진수 | 이재환 | 유은서 | 최수녕 | 이지은 |

# 내용공유

https://bit.ly/2YsxwEl

# 휴식 매시간 25분 ~ 35분점심 12시 20분 ~ 13시 30분

# 교육시간 09:00~18:00 (중식 : 12:00~13:00)

# 이름을 "OO조\_OOO" 형식(예: 1조\_홍길동)으로 바꿔 주세요.

# 출석, 퇴실은 해당 시간에 제시되는 QR로

# 출석체크 : 08:50~09:10 (이후 출석자는 매니저에게 밴드로 연락)

# 퇴실체크 : 17:50~18:10

# 매시간 정시에 QR 체크해야 합니다. (확인 버튼 누르고 QR 스캔)



# 8/31(월)~9/3(목) 인터페이스 개발 프로젝트

# 9/2 조별 리뷰 (프로그램 및 포트폴리오)

# 포트폴리오 양식은 밴드 공지 참조

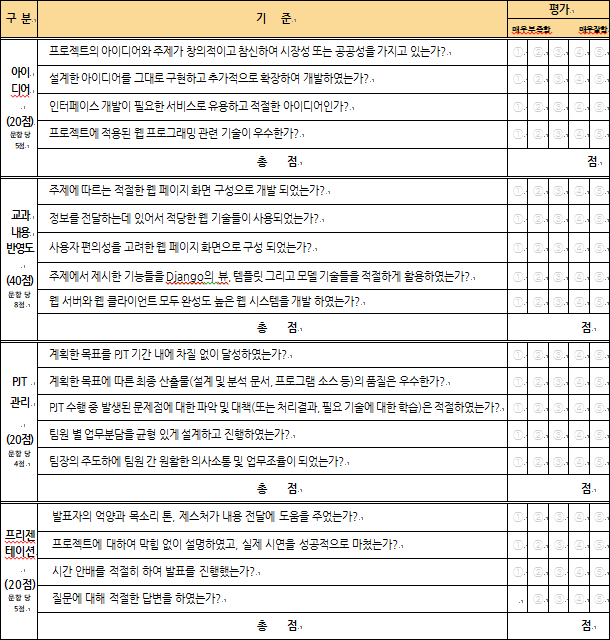
# 9/3 18:00 포트폴리오 제출 (밴드 참조)

# 

# 9/4(금) ~ 전공교과목 진행

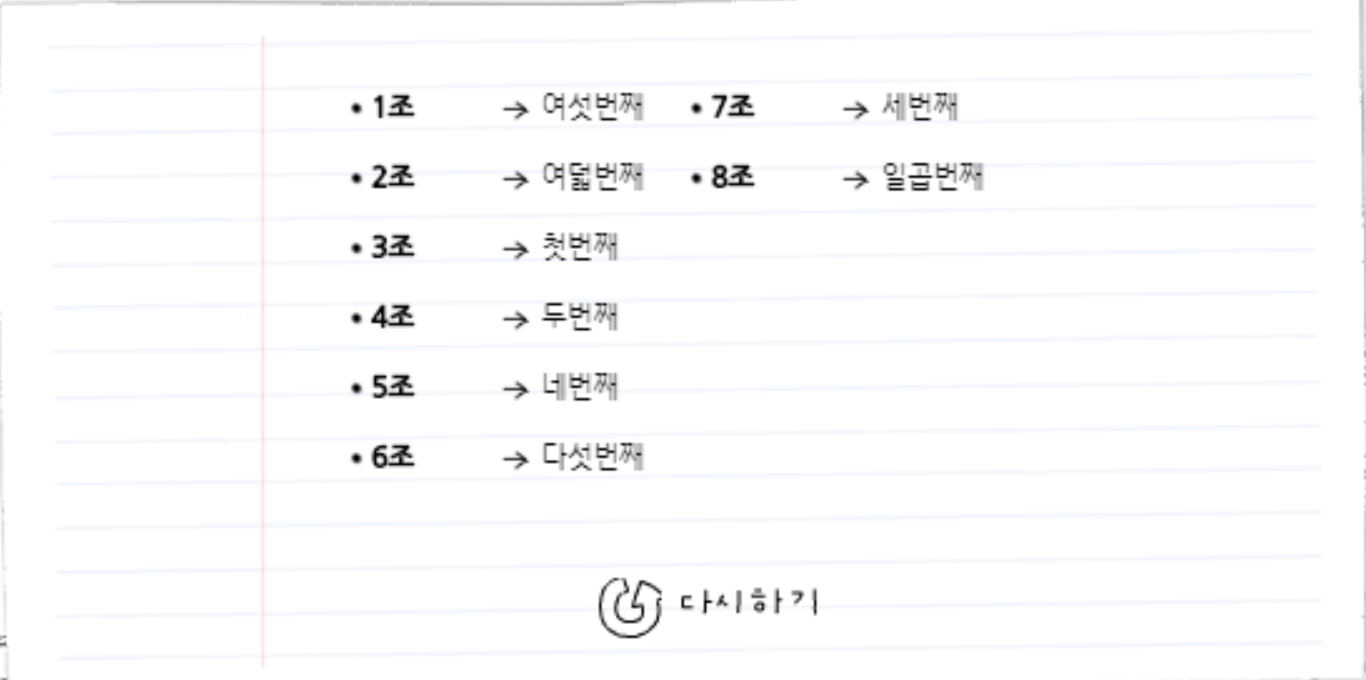
# 

인터페이스 개발 프로젝트 평가표



# 9/2 (수) 조별 리뷰

13시 30분부터 조별 30분 정도 리뷰를 하도록 하겠습니다.



**어제 보여줬던 다른 과정 발표자료 공유합니다. 참고하세요. 👀👀**

https://drive.google.com/drive/folders/1RIZKF2oF5NvQhnIGcO5n9Post4ea-BC2?usp=sharing

# 9/3 (목) 실습 환경 구성

## 포트폴리오 제출

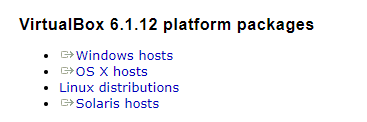
18시까지 cw1025.min@multicampus.com, myanjini@gmail.com 메일로 제출

## 

## 아래 설치 파일들을 미리 다운로드 해 주세요.

**#1 VirtualBox는 본인 PC 환경에 맞는 파일을 다운로드해야 합니다.**

<https://www.virtualbox.org/wiki/Downloads>



**#2 우분투 이미지 파일은 운영체제(윈도우, 맥)와 무관합니다.**

**#2-1 우분투 데스크탑**

http://old-releases.ubuntu.com/releases/18.04.4/ubuntu-18.04.4-desktop-amd64.iso

**#2-2 우분투 서버**

http://old-releases.ubuntu.com/releases/18.04.4/ubuntu-18.04.4-live-server-amd64.iso

**#2-3 쿠분투 데스크탑**

http://cdimage.ubuntu.com/kubuntu/releases/18.04/release/kubuntu-18.04.5-desktop-amd64.iso

**참고: 우분투, 쿠분투 다운로드 페이지**

* **Ubuntu** ⇒ <http://old-releases.ubuntu.com/releases/18.04.4/>
* **Kubuntu** ⇒ <https://kubuntu.org/getkubuntu/>

시스템 요구사항

<https://help.ubuntu.com/community/Installation/SystemRequirements>

<https://ubuntu.com/server/docs/installation>

VirtualBox 설치

<https://myanjini.tistory.com/entry/VirtualBox-%EC%84%A4%EC%B9%98>

Ubuntu Desktop 설치 (교재 62~67 페이지 참조)

<https://myanjini.tistory.com/entry/Ubuntu-Desktop-%EC%84%A4%EC%B9%98>

Ubuntu Desktop 초기 설정 (교재 68~77 페이지 참조)

<https://myanjini.tistory.com/entry/Ubuntu-Desktop-%EC%B4%88%EA%B8%B0-%EC%84%A4%EC%A0%95>

Ubuntu Server 설치 (교재 80~85 페이지 참조)

<https://myanjini.tistory.com/entry/Ubuntu-Server-%EC%84%A4%EC%B9%98>

Ubuntu Server 초기 설정 (교재 86~91 페이지 참조)

<https://myanjini.tistory.com/entry/Ubuntu-Server-%EC%B4%88%EA%B8%B0-%EC%84%A4%EC%A0%95>

9/4 (금)

## 용어

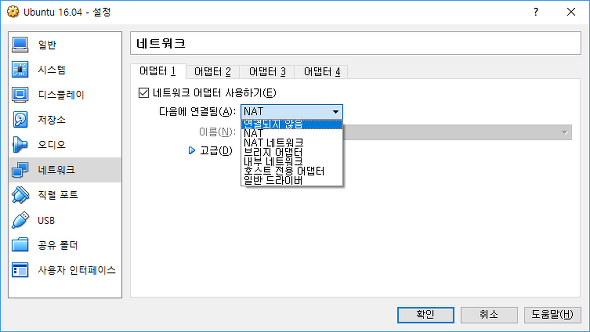
HOST - 가상화 소프트웨어가 동작하는 머신(PC)

~~~~~~~~~~~~~~~~~

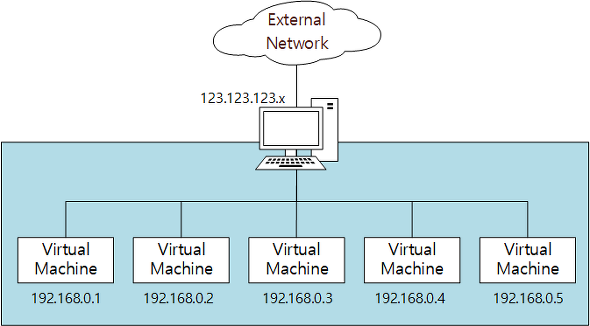
VMware Workstation/Player, VirtualBox, …

GUEST OS - 가상화 소프트웨어 안에서 동작하는 운영체제

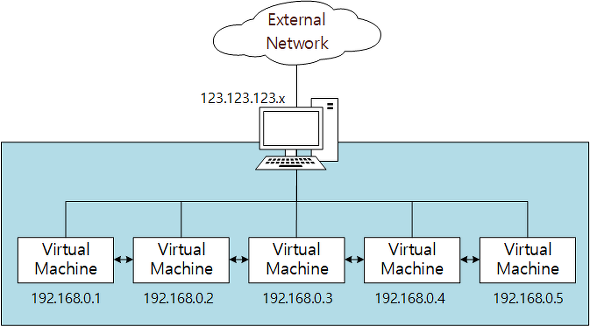
## VirtualBox Network 종류



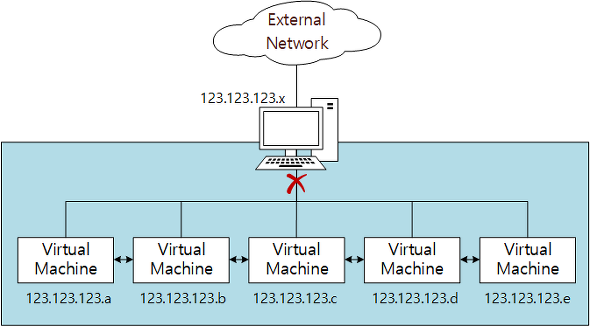
### NAT(Network Address Translation)



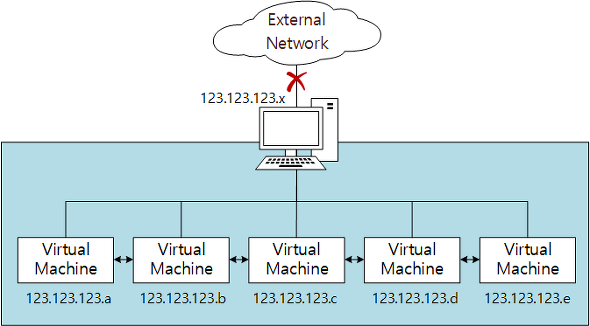
### NAT Network



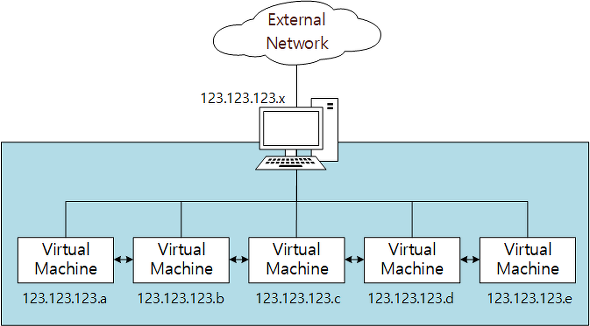
### Internal Network (내부 네트워크)



### Host-Only Adapter (호스트 전용)



### Bridge Adapter (브릿지)



## 잠시 쉬고, 10시 20분에 다시 시작하겠습니다.

## 저자가 만든 가상머신 이미지 가져와서 실행

#1 아래 사이트에서 2개의 파일을 다운로드합니다.

https://download.hanbit.co.kr/ubuntu/18.04/

[VMware-player-15.0.3-12422535.exe](https://drive.google.com/file/d/1BGLrgQ6kHWpUebz1gVmUUUYEBxSV0A1E/view?usp=drive_web)

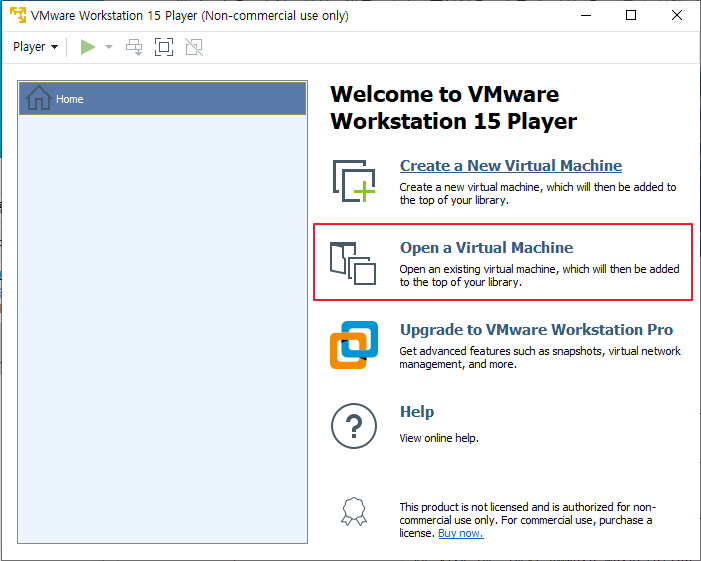
[Ubuntu18.04\_Server.exe](https://drive.google.com/file/d/1uMvtpdKADzY47ZpFLozZ6X5tOYvPxIyx/view?usp=drive_web)

#2 [VMware-player-15.0.3-12422535.exe](https://drive.google.com/file/d/1BGLrgQ6kHWpUebz1gVmUUUYEBxSV0A1E/view?usp=drive_web) 을 실행해서 설치를 진행합니다. (기본 설정 상태로 설치 진행)

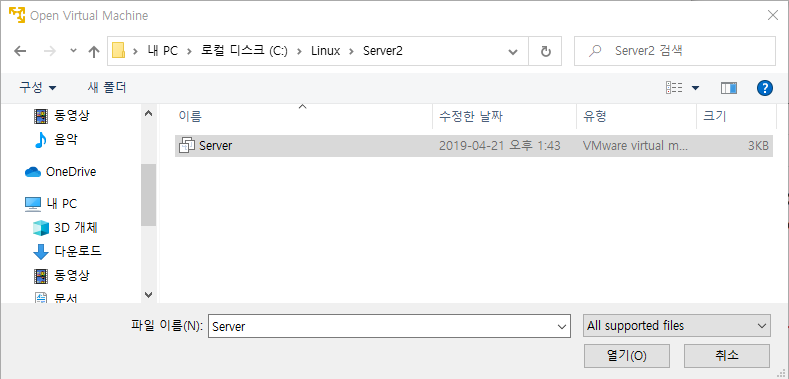
#3 [Ubuntu18.04\_Server.exe](https://drive.google.com/file/d/1uMvtpdKADzY47ZpFLozZ6X5tOYvPxIyx/view?usp=drive_web) 을 실행해서 압축해제합니다. (압축 해제할 폴더는 중복되지 않게 설정)

#4 시작 메뉴에서 VMware Workstation 15 Player 를 실행합니다.

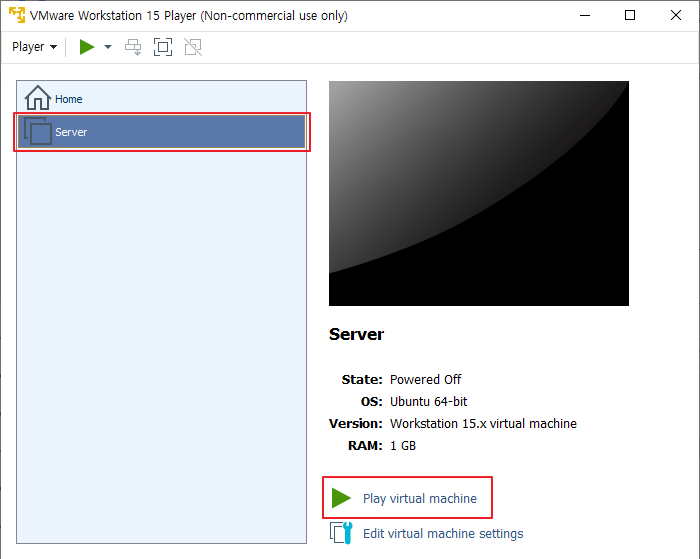
#5 만들어 놓은 가상 머신 이미지를 불러와서 실행



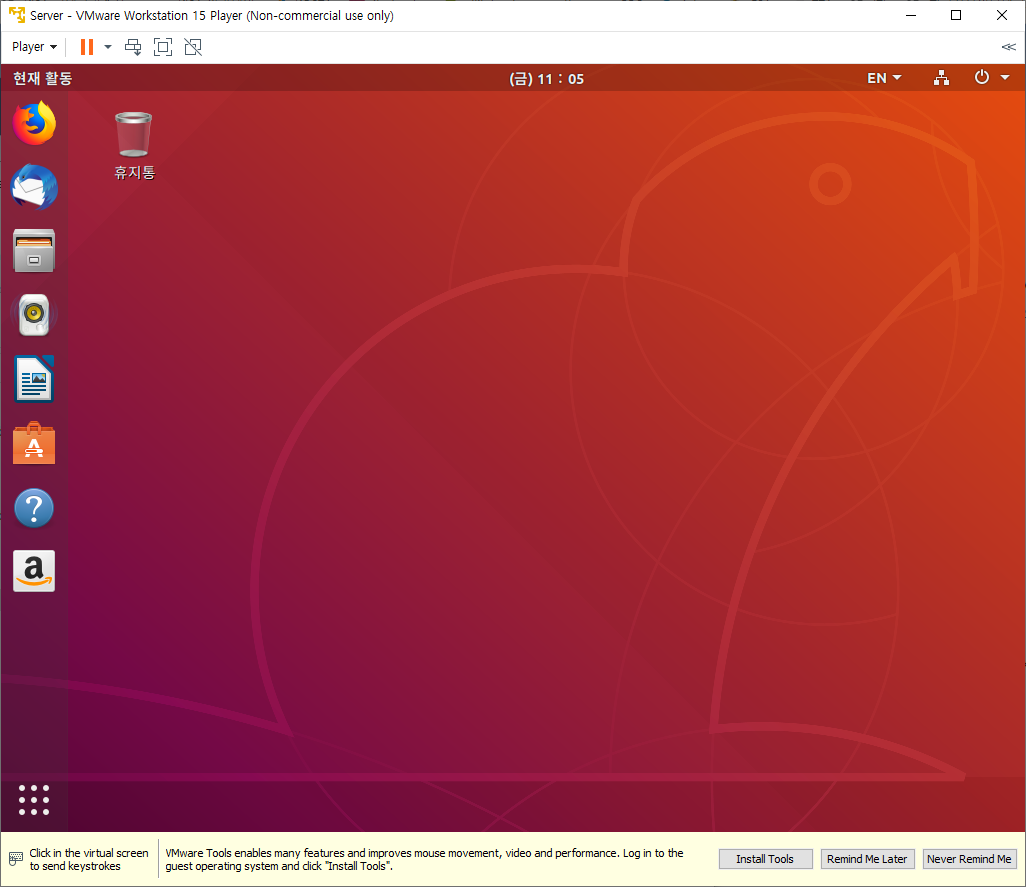
#6 #3에서 압축해제한 디렉터리에 server.vmx 파일을 선택



#7 교재의 저자가 만들어서 배포한 가상머신을 실행



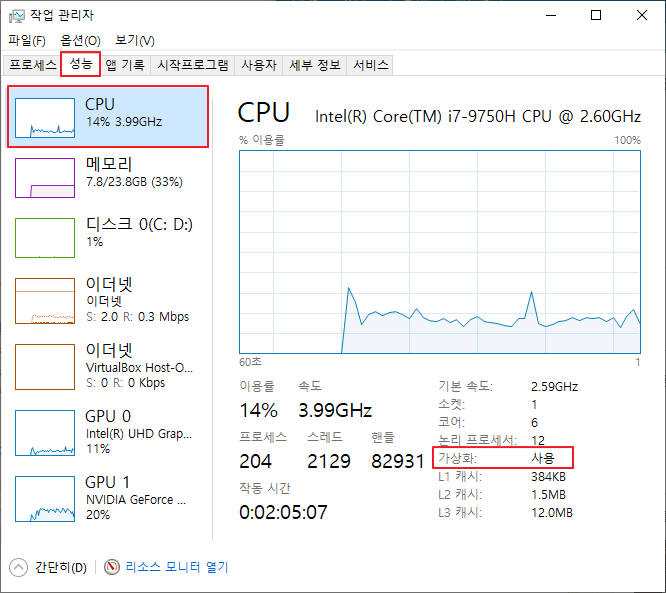
#8 우분투 실행을 확인



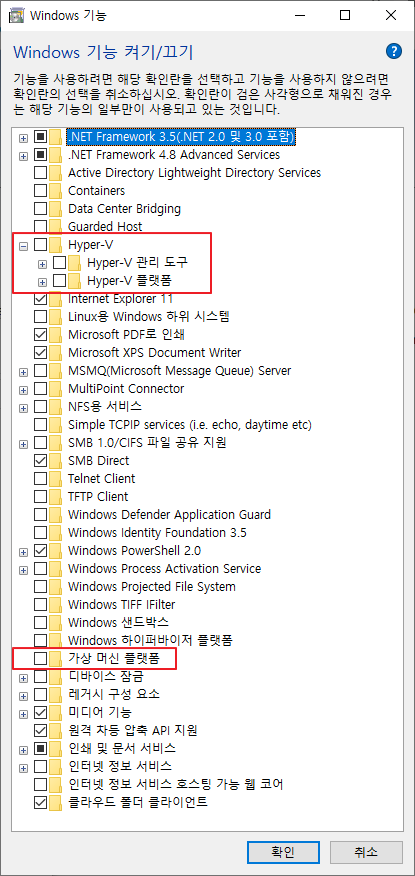
## VirtualBox 시작 시 오류가 나는 경우

### #1 가상화 여부 확인

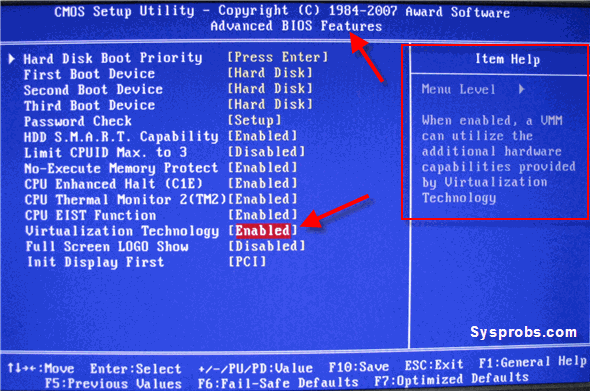
작업관리자(Ctrl+Alt+Del)

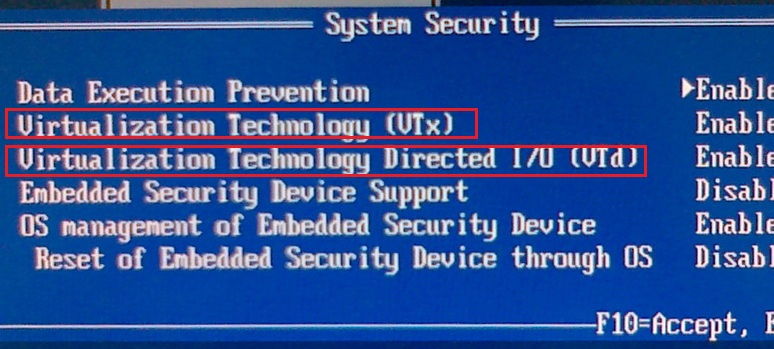
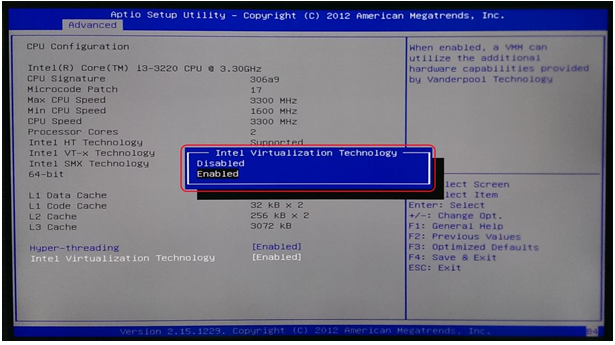


### #2 Windows 기능 켜기/끄기 설정 확인/변경



### #3 CMOS 가상화 설정(활성화)





## 11시 20분에 이어서 진행하겠습니다.

## 리눅스 커널 소스 확인

https://github.com/torvalds/linux

## 점심 식사 후 13시에 이어서 진행하겠습니다.

https://webinar.kafe.or.kr/room

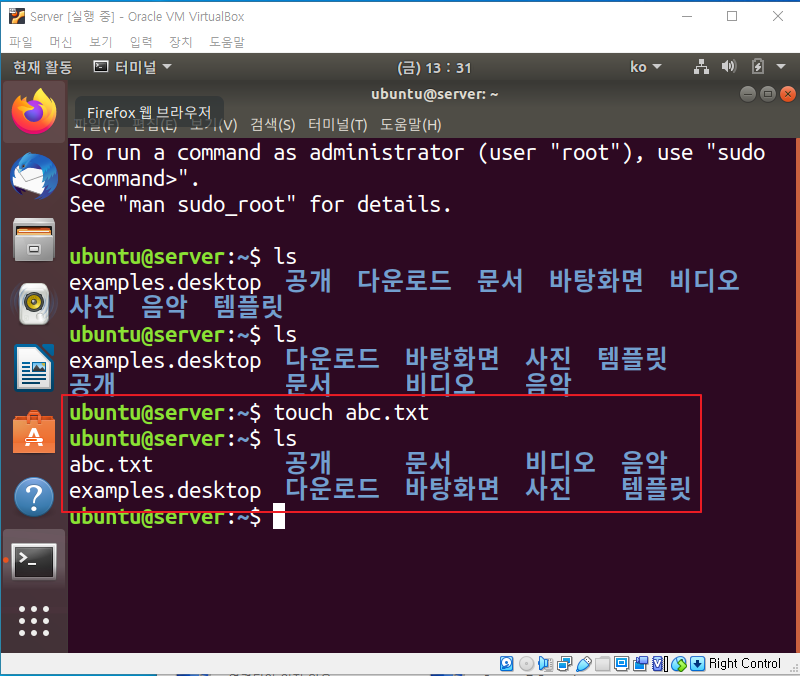
Seminar Name : 클라우드

JOIN

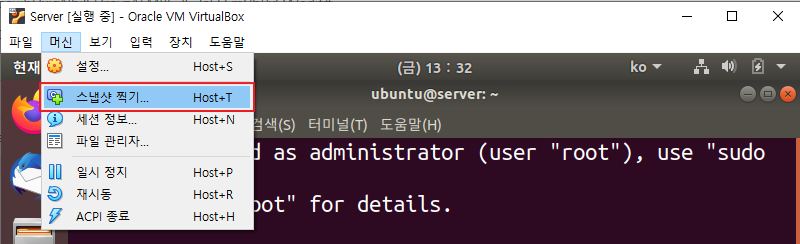
비밀번호 : zmffkdnem@123 (클라우드@123)

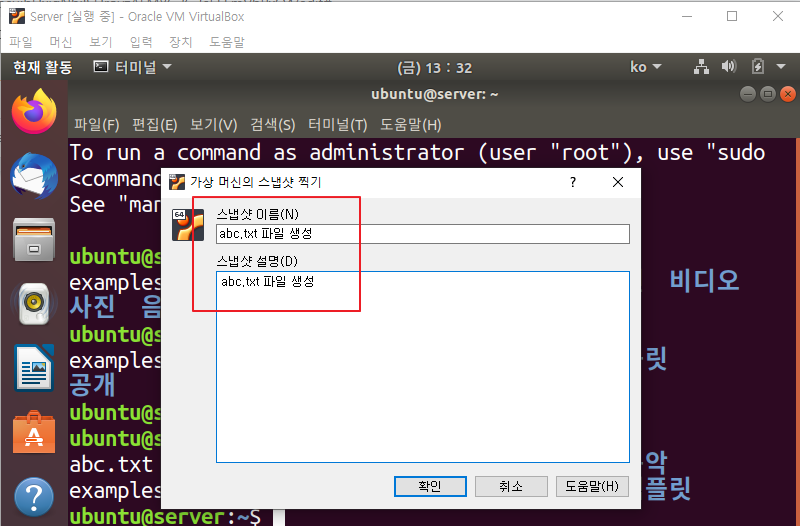
## VirtualBox 스냅샷 기능

### #1 터미널에서 abc.txt 파일을 생성

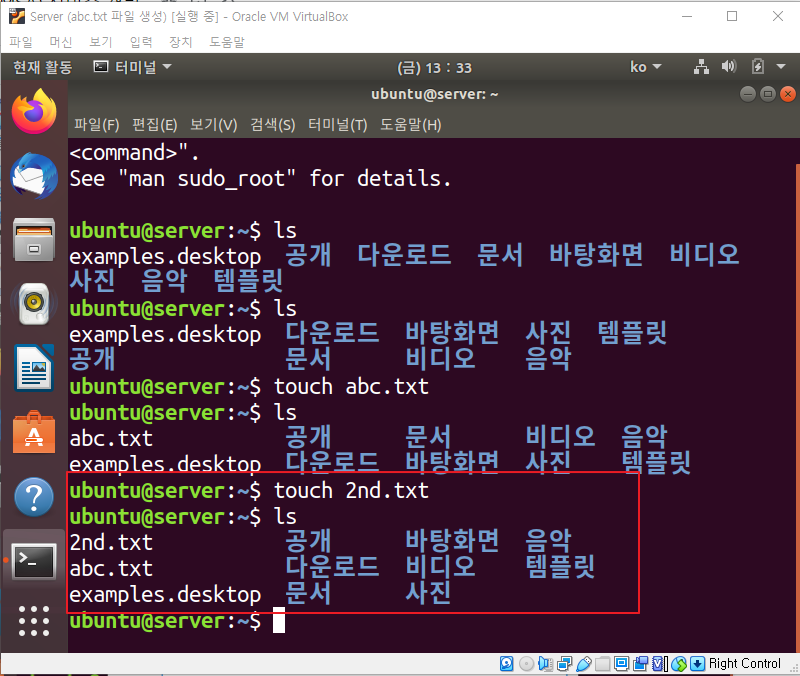


### #2 새로운 파일을 생성한 상태를 저장

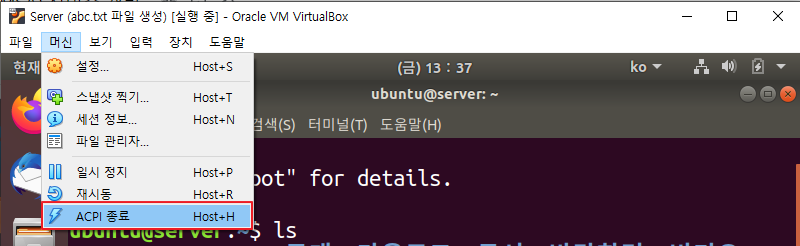




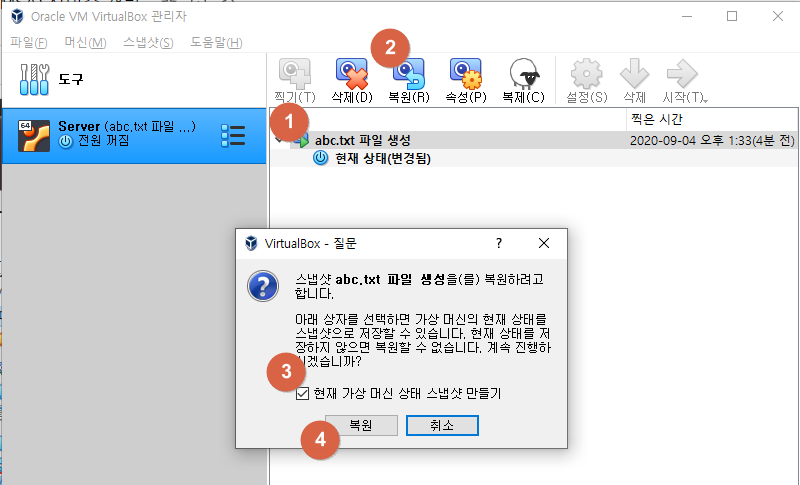
### #3 새로운 파일을 추가

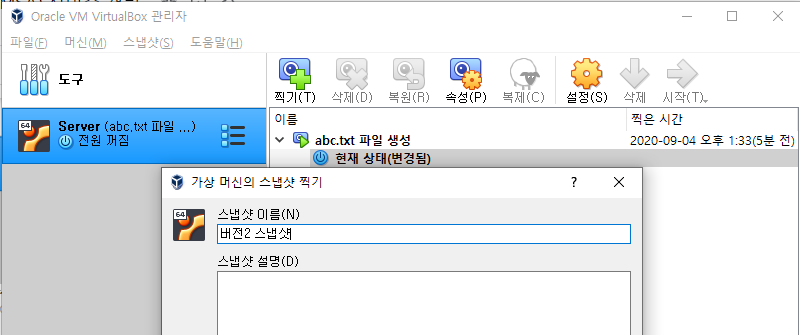


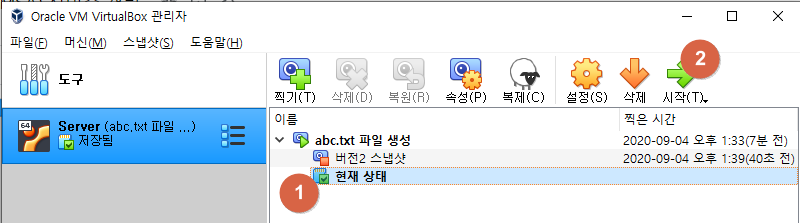
### #4 #2 상태로 되돌아 가고 싶은 경우

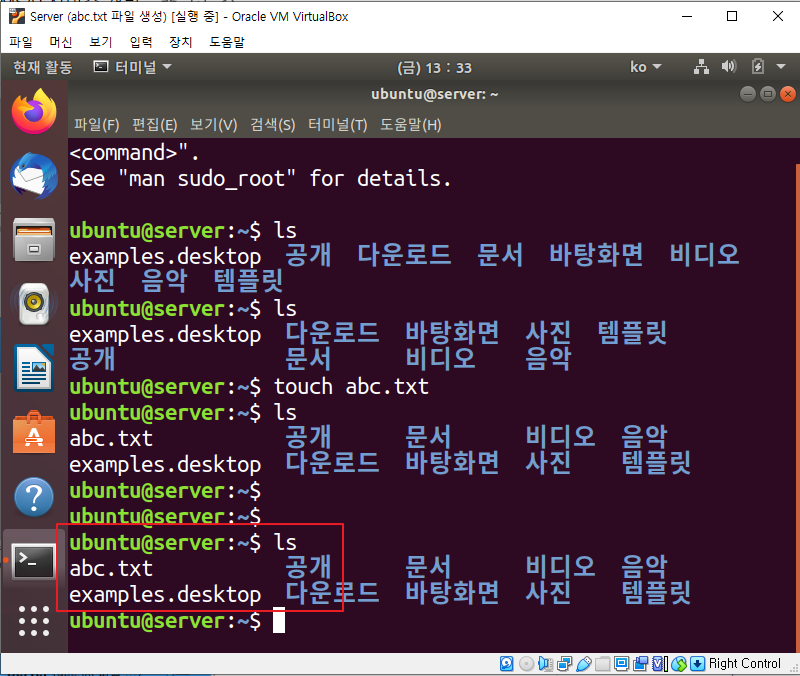




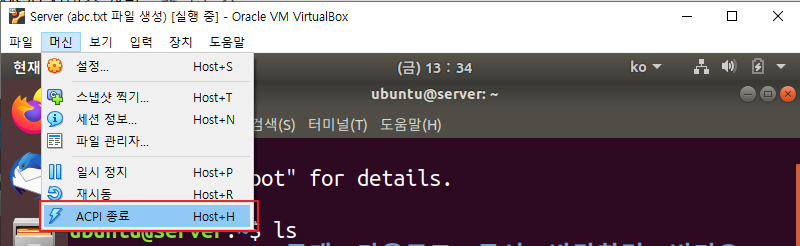


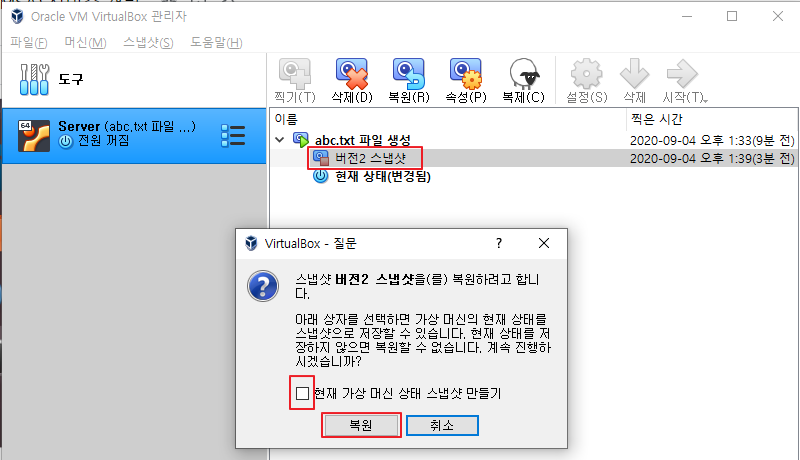


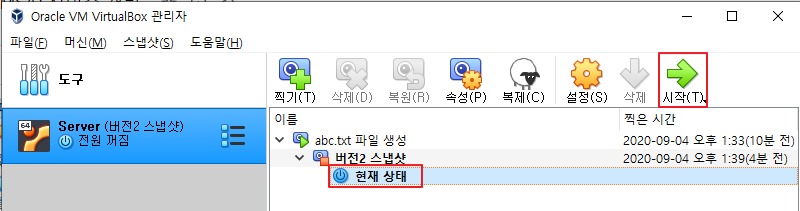




### #5 #3번 상태로 이동(복원)

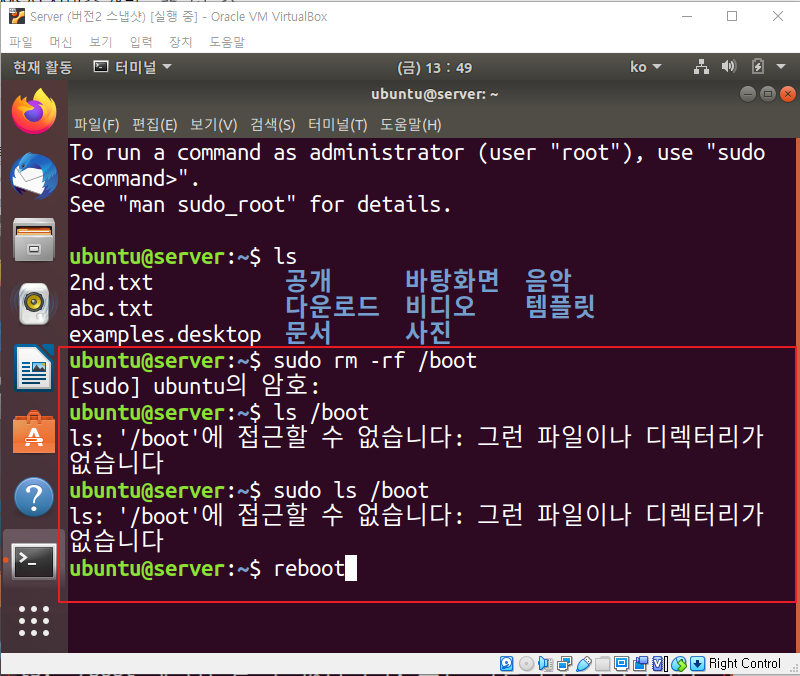




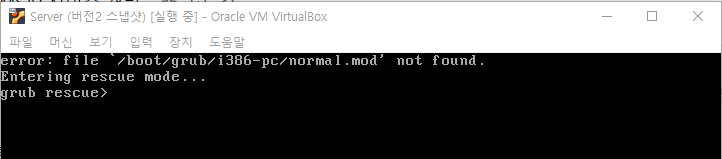




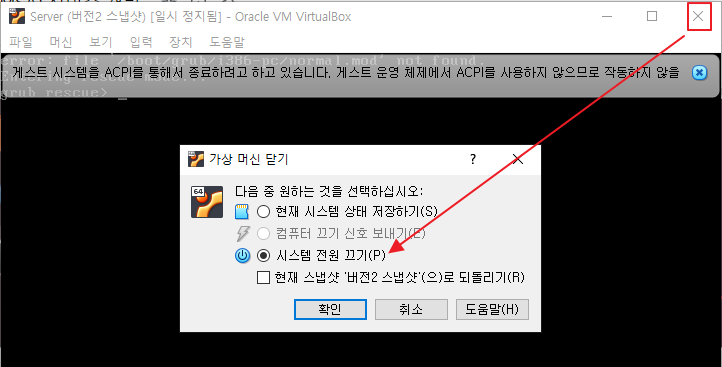
### #6 실수로 시스템 파일을 삭제 후 리부팅

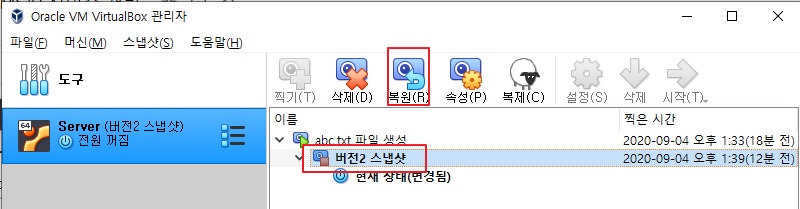


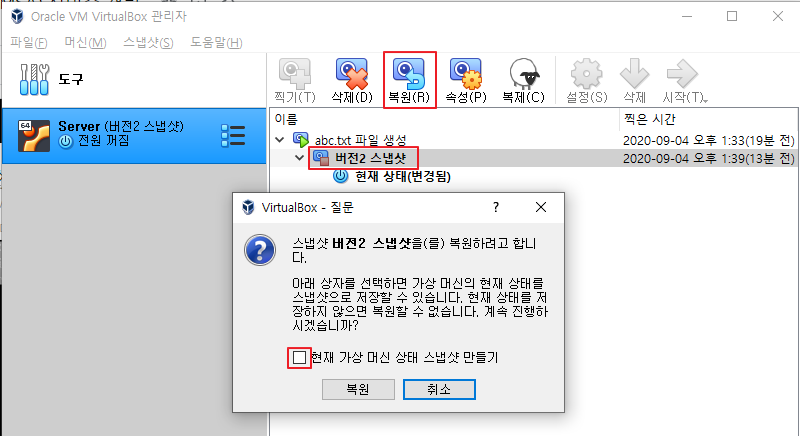
### #7 부팅이 되지 않는 것을 확인



### #8 스냅샷을 이용해서 복구





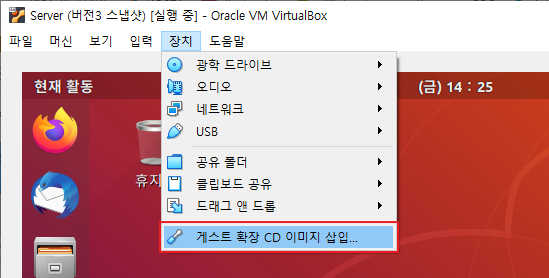


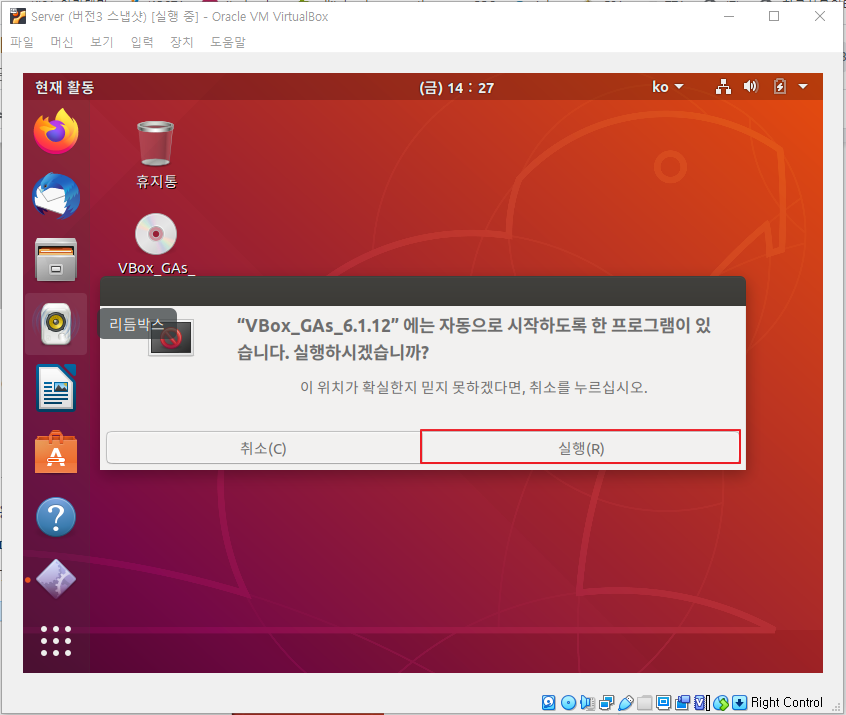
## 잠시 쉬고, 14시 20분에 이어서 진행하겠습니다.

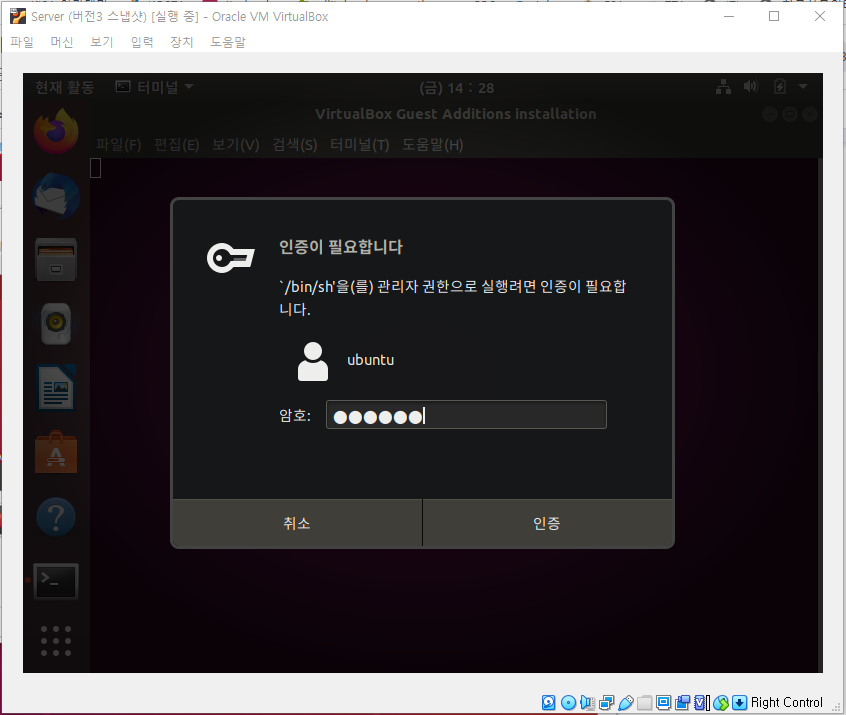
줌으로 다시 접속해 주세요.

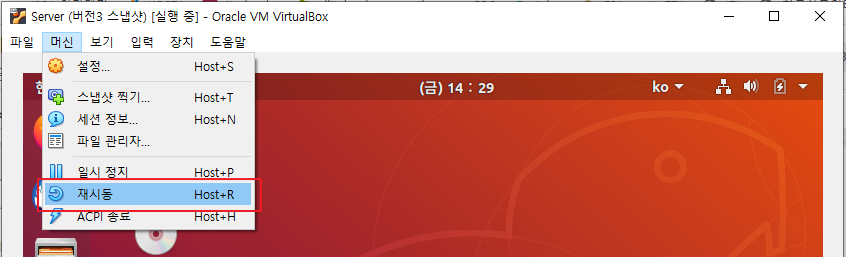
## VirtualBox 확장 기능 이용

### #1 장치 > 게시트 확장 CD 이미지 삽입







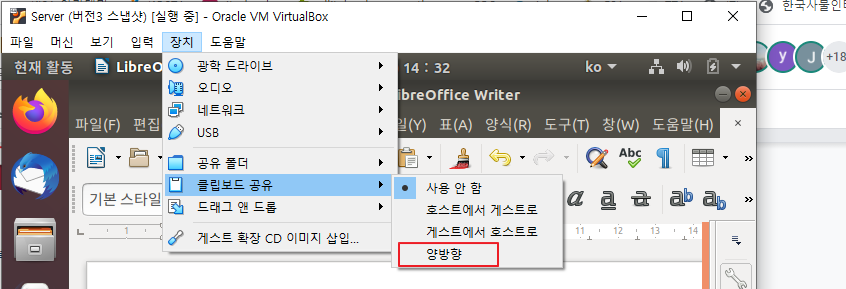


### #2 재시동 후 윈도우를 확대, 축소했을 때 해상도가 변경되는 것을 확인

### 

### #3 호스트와 게스트 간 클립보드 공유를 설정

### #3-1 장치 > 클립보드 공유 > 양방향을 선택 후 리부팅



### 

### #3-2 호스트와 게스트 간 Copy & Paste 동작 여부를 확인

### #4 호스트와 게스트 간 공유 폴더 설정

https://gist.github.com/estorgio/0c76e29c0439e683caca694f338d4003

### #4-1 호스트 머신에 공유에 사용할 디렉터리를 생성

### #4-2 root 사용자로 로그인 후 게스트 머신에 공유에 사용할 디렉터리를 생성

사용자 머신이름

~~~~~~ ~~~~~~

ubuntu@server:~$ sudo su

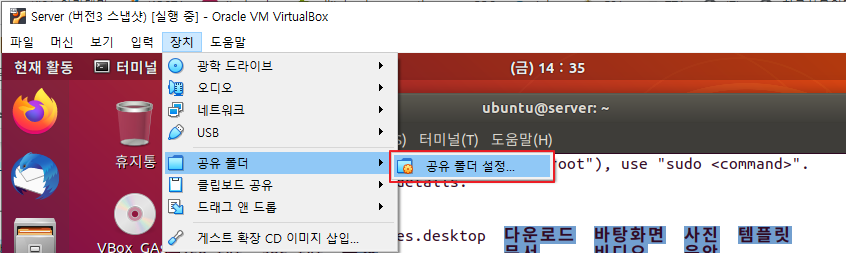
root@server:/home/ubuntu# cd

root@server:~# mkdir /ShareFolder

# cd ⇐ 루트 사용자가 cd 명령어를 실행

$ cd ⇐ 일반 사용자가 cd 명령어를 실행

### #4-3 장치 > 공유 폴더 > 공유 폴더 설정을 설정



### #4-4 설정

### 

폴더 경로 : 호스트 머신에서 사용할 공유 폴더 경로

폴더 이름 : 폴더를 구분하기 위한 이름

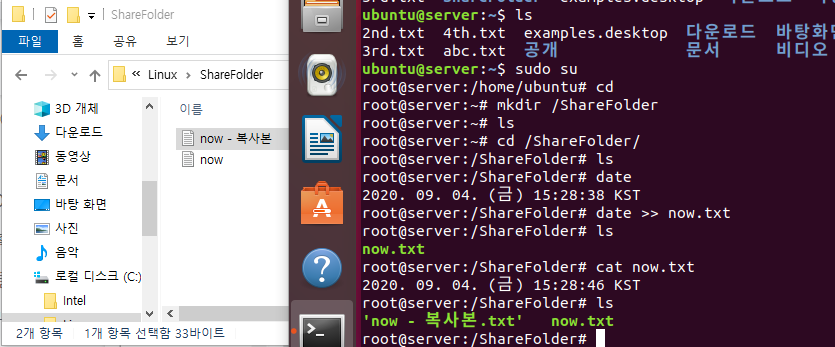
읽기 전용 : 게스트 운영체제에서 읽기 용도로만 사용 (선택하지 않으면 읽고 쓸 수 있음)

자동 마운트 : 게스트 운영체제가 실행될 때 자동으로 공유가 되도록 설정

마운트 지정 : 게스트 머신에서 사용할 공유 폴더 경로 (/ShareFolder)

### #4-5 파일 공유를 확인

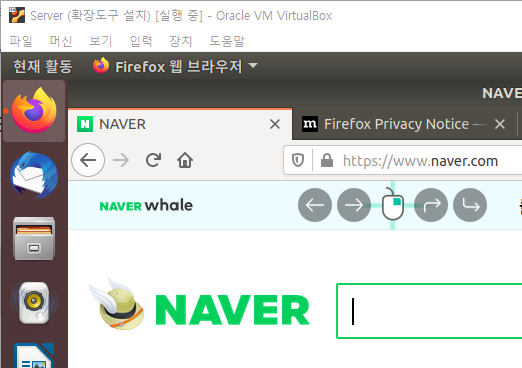




## 잠시 쉬고, 15시 20분에 이어서 진행하겠습니다.

## 포트 포워딩을 이용한 내부 네트워크 이용

### #1 FireFox 브라우저를 실행해서 네이버로 접속되는 것을 확인



### #2 우분투에 nginx 서버를 설치

ubuntu@server:~$ sudo apt-get update

[sudo] ubuntu의 암호: *ubuntu*

받기:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]

기존:2 http://kr.archive.ubuntu.com/ubuntu bionic InRelease

받기:3 http://kr.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]

받기:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metadata [48.9 kB]

받기:5 http://kr.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]

받기:6 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Metadata [56.0 kB]

받기:7 http://security.ubuntu.com/ubuntu bionic-security/universe DEP-11 64x64 Icons [109 kB]

받기:8 http://kr.archive.ubuntu.com/ubuntu bionic

:

ubuntu@server:~$ sudo apt-get install -y nginx

패키지 목록을 읽는 중입니다... 완료

의존성 트리를 만드는 중입니다

상태 정보를 읽는 중입니다... 완료

다음의 추가 패키지가 설치될 것입니다 :

libnginx-mod-http-geoip libnginx-mod-http-image-filter

libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream

nginx-common nginx-core

제안하는 패키지:

fcgiwrap nginx-doc

다음 새 패키지를 설치할 것입니다:

libnginx-mod-http-geoip libnginx-mod-http-image-filter

:

ubuntu@server:~$ sudo service nginx restart

ubuntu@server:~$ sudo service nginx status

● nginx.service - A high performance web server and a reverse proxy server

Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: en

Active: active (running) since Fri 2020-09-04 15:43:21 KST; 8s ago

Docs: man:nginx(8)

Process: 3448 ExecStop=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 -

Process: 3450 ExecStart=/usr/sbin/nginx -g daemon on; master\_process on; (code

Process: 3449 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master\_process

Main PID: 3451 (nginx)

Tasks: 2 (limit: 2328)

CGroup: /system.slice/nginx.service

├─3451 nginx: master process /usr/sbin/nginx -g daemon on; master\_pro

└─3452 nginx: worker process

9월 04 15:43:21 server systemd[1]: Stopped A high performance web server and a

:

### #3 우분투 가상머신의 IP를 확인

ubuntu@server:~$ ip a

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: **enp0s3**: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000

link/ether 08:00:27:96:b2:75 brd ff:ff:ff:ff:ff:ff

inet **10.0.2.4**/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3

valid\_lft 525sec preferred\_lft 525sec

inet6 fe80::2e79:698a:dba3:9968/64 scope link noprefixroute

valid\_lft forever preferred\_lft forever

### #4 우분투에서 nginx로 접속

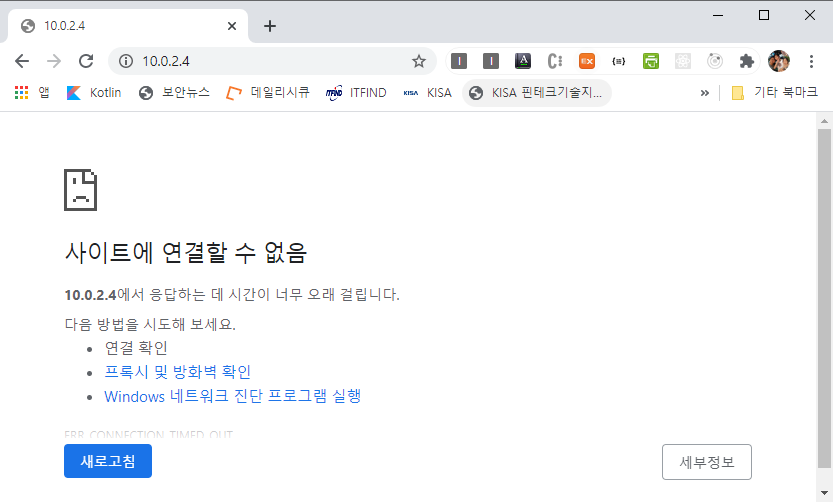
http://localhost

http://127.0.0.1

http://**10.0.2.4**



### #5 호스트 머신에서 우분투 머신의 IP로 nginx에 접속되는지 확인



### #6 호스트 머신의 IP 주소를 확인

Microsoft Windows [Version 10.0.18363.753]

(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\myanj> ipconfig

Windows IP 구성

**이더넷 어댑터 이더넷**: ⇐ 랜카드 = **NIC**(Network Interface Controller)

~~~~~~ ⇒ LAN 영역에서 사용하는 통신기술 중 하나

LAN 영역에서 사용하는 기술 중 사실상 표준(De Factor Standard) 방식

연결별 DNS 접미사. . . . :

링크-로컬 IPv6 주소 . . . . : fe80::467:328a:16f3:43f1%6

IPv4 주소 . . . . . . . . . : 192.168.0.218 ⇐ 32비트로 구성된 주소 체계 (232 개의 주소를 표현)

~~ 0.0.0.0~255.255.255.255

서브넷 마스크 . . . . . . . : 255.255.255.0

기본 게이트웨이 . . . . . . : 192.168.0.1 ⇐ 게이트웨이 → SW 측면 / 라우터 → HW 측명

각기 다른 네트워크ID를 사용하는 LAN 영역을 연결해주는 기능

이더넷 어댑터 VirtualBox Host-Only Network: 스위칭(switching) ⇒ LAN 영역에서 내부 통신을 수행 (MAC 주소)

연결별 DNS 접미사. . . . :

링크-로컬 IPv6 주소 . . . . : fe80::d9df:db0e:379f:7e13%41

IPv4 주소 . . . . . . . . . : 192.168.56.1

서브넷 마스크 . . . . . . . : 255.255.255.0

기본 게이트웨이 . . . . . . :

무선 LAN 어댑터 Wi-Fi:

미디어 상태 . . . . . . . . : 미디어 연결 끊김

연결별 DNS 접미사. . . . :

IP(Internet Protocol) 주소

인터넷 공간에서 자기 PC가 사용하는 고유한 식별자를 의미

서브넷 마스크(subnet mask)

IP 주소에서 네트워크 ID와 호스트 ID를 구분하는 방법

IPv4 주소 . . . . . . . . . : 192.168. 0.218

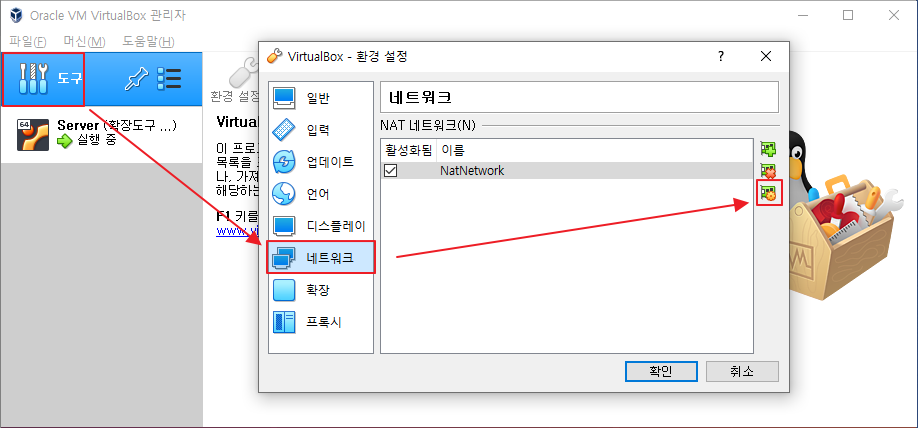
서브넷 마스크 . . . . . . . : 255.255.255.0

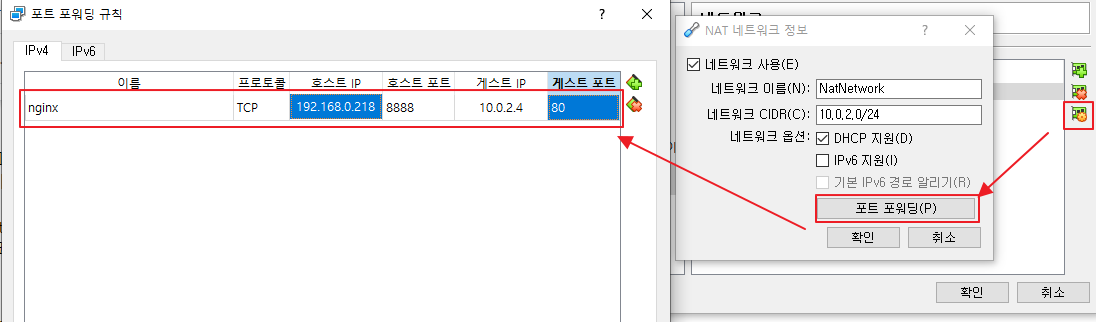
~~~~~~~~~~~ ~~~

192.168.0 218

네트워크ID 호스트ID

### #7 NatNetwork 수정





호스트 IP : 호스트 머신에서 사용하는 IP 주소

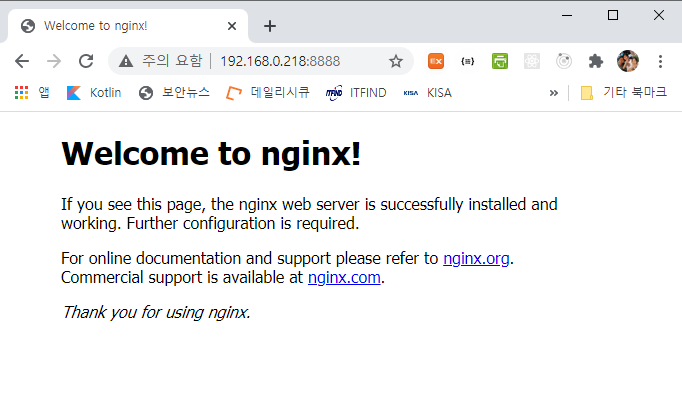
호스트 포트 : 호스트 머신에서 포트 포워딩에 사용할 포트 번호

게스트 IP : 게스트 머신(우분트)에서 사용하고 있는 IP 주소

게스트 포트 : 게스트 머신에서 사용하고 있는 서비스 포트

### #8 호스트 머신에서 접속 확인

http://192.168.0.218:8888



## 잠시 쉬고, 16시 50분에 이어서 진행하겠습니다.

### 우분투 서버, 우분투 데스크탑, 쿠분투 설치를 꼭 해 보세요. (106페이지)

Ubuntu Desktop 설치 (교재 62~67 페이지 참조)

<https://myanjini.tistory.com/entry/Ubuntu-Desktop-%EC%84%A4%EC%B9%98>

Ubuntu Desktop 초기 설정 (교재 68~77 페이지 참조)

<https://myanjini.tistory.com/entry/Ubuntu-Desktop-%EC%B4%88%EA%B8%B0-%EC%84%A4%EC%A0%95>

Ubuntu Server 설치 (교재 80~85 페이지 참조)

<https://myanjini.tistory.com/entry/Ubuntu-Server-%EC%84%A4%EC%B9%98>

Ubuntu Server 초기 설정 (교재 86~91 페이지 참조)

<https://myanjini.tistory.com/entry/Ubuntu-Server-%EC%B4%88%EA%B8%B0-%EC%84%A4%EC%A0%95>

### vi 에디터 사용법을 연습해 보세요. (121 ~ 128 페이지)

9/7 (월)

## 온라인 강의 접속 URL

<https://www.myroom.hpe.com/attend/MRP2PY97CYKKU>

아래 공유 폴더에 조별 포트폴리오를 공유해 뒀습니다.

다른 조는 어떻게 했는지 벤치마킹해 보고 더 좋은 결과물을 만드는데 활용해 보세요.

https://drive.google.com/drive/folders/1RIZKF2oF5NvQhnIGcO5n9Post4ea-BC2?usp=sharing

## Bitvise SSH Client Download

https://www.bitvise.com/ssh-client-download ⇐ 다운로드 페이지

https://dl.bitvise.com/BvSshClient-Inst.exe ⇐ 설치 파일 다운로드

Client <-----------> Ubuntu Server

SSH Client SSH Server

~~~~~~~~~~ Putty

### #1 Ubuntu Desktop IP 확인

ubuntu@server:~$ ip a

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000

link/ether 08:00:27:96:b2:75 brd ff:ff:ff:ff:ff:ff

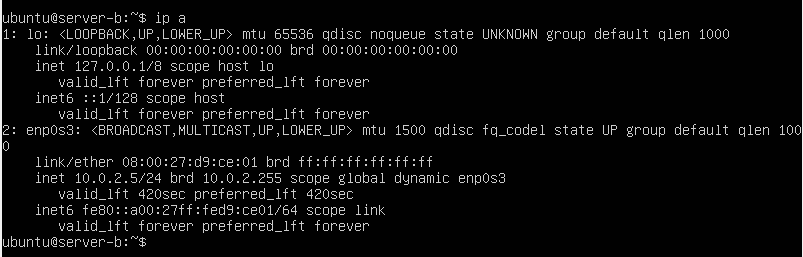
inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3

valid\_lft 507sec preferred\_lft 507sec

inet6 fe80::2e79:698a:dba3:9968/64 scope link noprefixroute

valid\_lft forever preferred\_lft forever

### #2 Ubuntu Server IP 확인



### #3 Ubuntu Desktop에서 Ubuntu Server로 SSH 접속

ubuntu@server:~$ ssh 10.0.2.5 ⇐ Ubuntu Server IP

The authenticity of host '10.0.2.5 (10.0.2.5)' can't be established.

ECDSA key fingerprint is SHA256:XsaQWtyiPJhtxTjcUYaMF3sFuYDq2WzSanz4HPEf5DY.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '10.0.2.5' (ECDSA) to the list of known hosts.

ubuntu@10.0.2.5's password: ubuntu ⇐ Ubuntu Server Password

Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-115-generic x86\_64)

\* Documentation: https://help.ubuntu.com

\* Management: https://landscape.canonical.com

\* Support: https://ubuntu.com/advantage

System information as of Mon Sep 7 00:17:10 UTC 2020

System load: 0.0 Processes: 87

Usage of /: 20.2% of 19.56GB Users logged in: 1

Memory usage: 8% IP address for enp0s3: 10.0.2.5

Swap usage: 0%

65 packages can be updated.

0 updates are security updates.

Last login: Mon Sep 7 00:01:19 2020

ubuntu@server-b:~$ ip a

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000

link/ether 08:00:27:d9:ce:01 brd ff:ff:ff:ff:ff:ff

inet 10.0.2.5/24 brd 10.0.2.255 scope global dynamic enp0s3

valid\_lft 427sec preferred\_lft 427sec

inet6 fe80::a00:27ff:fed9:ce01/64 scope link

valid\_lft forever preferred\_lft forever

ubuntu@server-b:~$ hostname

server-b

### #4 Ubuntu Desktop에 SSH Server를 설치

$ sudo su

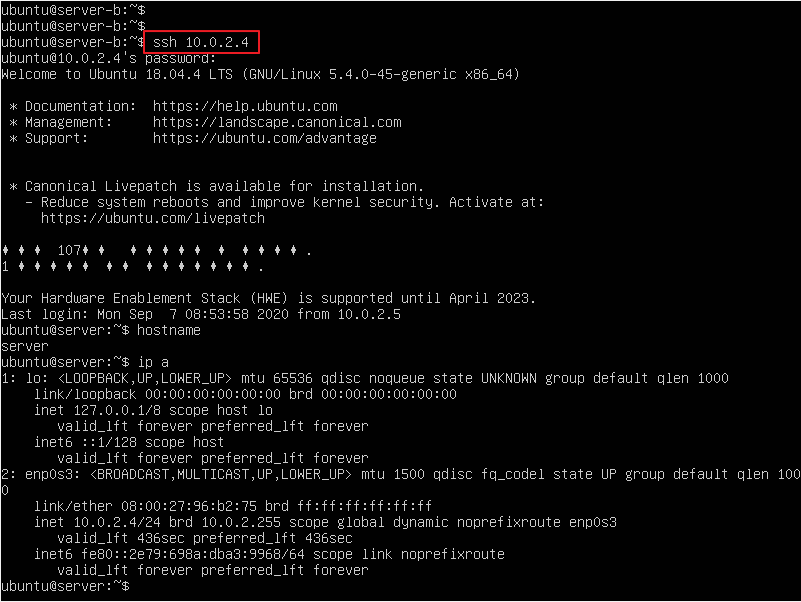
# apt update

# apt upgrade

# apt install openssh-server

|  |
| --- |
| 아래 오류가 발생하는 경우  openssh-server : 의존: openssh-client (= 1:7.2p2-4)  의존: openssh-sftp-server 하지만 %s 패키지를 설치하지 않을 것입니다  추천: ssh-import-id 하지만 %s 패키지를 설치하지 않을 것입니다  # apt install openssh-client=1:7.2p2-4  # apt install openssh-server |

### #5 Ubuntu Server에서 Ubuntu Desktop으로 SSH 접속



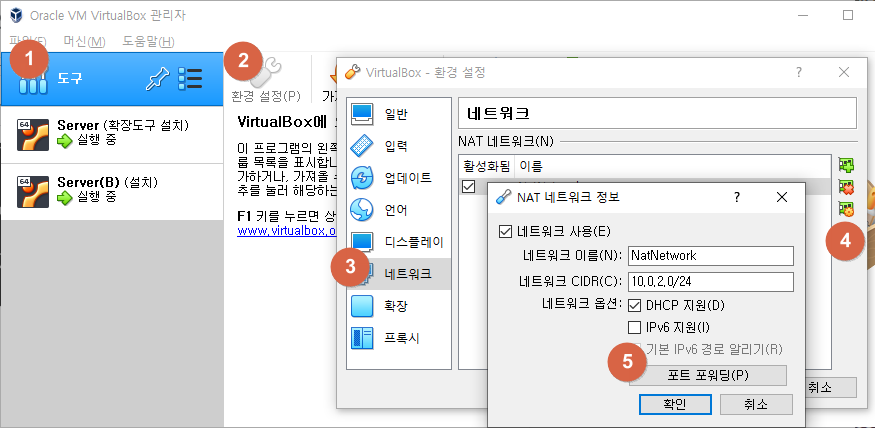
### #6 Host PC에서 Ubuntu Desktop으로 SSH 접속

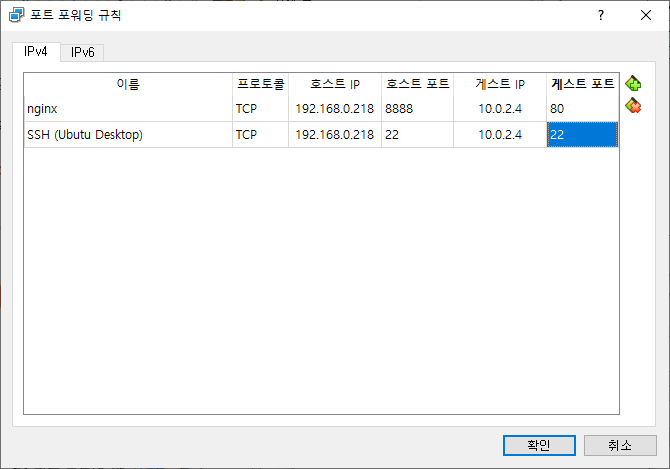
### #6-1 명령어창을 실행한 후 Ubuntu Desktop으로 접속

C:\Users\myanj> ssh 10.0.2.4

ssh: connect to host 10.0.2.4 port 22: Connection timed out

### #6-2 포트 포워딩 설정





* 호스트 IP : Host PC (자기 PC)의 IP
* 호스트 포트 : Host PC에서 사용하는 포트 번호
* 게스트 IP : 서비스를 제공할 가상 머신의 IP
* 게시트 포트 : 가상 머신에서 서비스를 제공에 사용하는 포트 번호

C:\Users\myanj> ssh 192.168.0.218 -p 22 ⇒ 명령어 오류 발생 시 https://charging.tistory.com/94 참조

The authenticity of host '192.168.0.218 (192.168.0.218)' can't be established.

ECDSA key fingerprint is SHA256:3yvF5H+nWFHo9yvJPrHT20AA7DErN5w+g6WXN7KaUIg.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.0.218' (ECDSA) to the list of known hosts.

myanj@192.168.0.218's password: ⇐ 접속 시 사용자를 지정하지 않으면 호스트 PC의 사용자로 접속

C:\Users\myanj> ssh ubuntu@192.168.0.218 -p 22 ⇐ ubuntu 사용자로 접속

ubuntu@192.168.0.218's password: ubuntu

Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-45-generic x86\_64)

\* Documentation: https://help.ubuntu.com

\* Management: https://landscape.canonical.com

\* Support: https://ubuntu.com/advantage

\* Canonical Livepatch is available for installation.

- Reduce system reboots and improve kernel security. Activate at:

https://ubuntu.com/livepatch

패키지 0개를 업데이트할 수 있습니다.

0 업데이트는 보안 업데이트입니다.

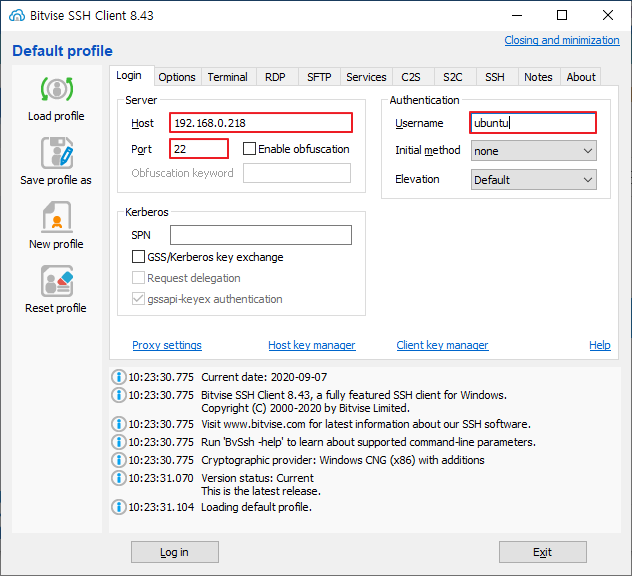
Your Hardware Enablement Stack (HWE) is supported until April 2023.

\*\*\* 시스템을 다시 시작해야 합니다 \*\*\*

Last login: Mon Sep 7 09:27:08 2020 from 10.0.2.5

ubuntu@server:~$ ⇐ Ubuntu Desktop으로 접속

### #6-3 Bitvise SSH Client를 이용해서 접속



Last login: Mon Sep 7 10:33:31 2020 from 192.168.0.218

ubuntu@server:~$ ip a

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000

link/ether 08:00:27:96:b2:75 brd ff:ff:ff:ff:ff:ff

inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3

valid\_lft 380sec preferred\_lft 380sec

inet6 fe80::2e79:698a:dba3:9968/64 scope link noprefixroute

valid\_lft forever preferred\_lft forever

ubuntu@server:~$

### #7 호스트 PC에서 Ubuntu Server로 SSH 접속

### #7-1 포트 포워딩 정보를 추가



### #7-2 SSH 접속

C:\Users\myanj> ssh ubuntu@192.168.0.218 -p 222

The authenticity of host '[192.168.0.218]:222 ([192.168.0.218]:222)' can't be established.

ECDSA key fingerprint is SHA256:XsaQWtyiPJhtxTjcUYaMF3sFuYDq2WzSanz4HPEf5DY.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '[192.168.0.218]:222' (ECDSA) to the list of known hosts.

ubuntu@192.168.0.218's password: ubuntu

Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-115-generic x86\_64)Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-115-generic x86\_64)

\* Documentation: https://help.ubuntu.com

\* Management: https://landscape.canonical.com

\* Support: https://ubuntu.com/advantage

System information as of Mon Sep 7 02:04:13 UTC 2020

System load: 0.0 Processes: 86

Usage of /: 20.2% of 19.56GB Users logged in: 1

Memory usage: 8% IP address for enp0s3: 10.0.2.5

Swap usage: 0%

\* Kubernetes 1.19 is out! Get it in one command with:

sudo snap install microk8s --channel=1.19 --classic

https://microk8s.io/ has docs and details.

65 packages can be updated.

0 updates are security updates.

Last login: Mon Sep 7 00:22:51 2020 from 10.0.2.4

ubuntu@server-b:~$

## 리눅스 기본 명령어

3장 ⇒ https://docs.google.com/presentation/d/1RwiwiFaWL9ovO4Kl9WOhnXDJKUdF0QlnqrJy3TSpujg/edit?usp=sharing

4장 ⇒ https://docs.google.com/presentation/d/1OX47wMg9AzOov3VdEvF6u1BIoBV5B45tKahPFL2361Y/edit?usp=sharing

## 런레벨 확인 및 변경

### 런레벨 파일 확인

root@server:~# ls -al /lib/systemd/system/runlevel?.target

lrwxrwxrwx 1 root root 15 7월 9 03:59 /lib/systemd/system/runlevel0.target -> poweroff.target

lrwxrwxrwx 1 root root 13 7월 9 03:59 /lib/systemd/system/runlevel1.target -> rescue.target

lrwxrwxrwx 1 root root 17 7월 9 03:59 /lib/systemd/system/runlevel2.target -> multi-user.target

lrwxrwxrwx 1 root root 17 7월 9 03:59 /lib/systemd/system/runlevel3.target -> multi-user.target

lrwxrwxrwx 1 root root 17 7월 9 03:59 /lib/systemd/system/runlevel4.target -> multi-user.target

lrwxrwxrwx 1 root root 16 7월 9 03:59 /lib/systemd/system/runlevel5.target -> graphical.target

lrwxrwxrwx 1 root root 13 7월 9 03:59 /lib/systemd/system/runlevel6.target -> reboot.target

root@server:~# ls -al /lib/systemd/system/default.target

lrwxrwxrwx 1 root root 16 7월 9 03:59 /lib/systemd/system/default.target -> graphical.target

### default.target을 변경

root@server:~# ls -al /lib/systemd/system/default.target

lrwxrwxrwx 1 root root 16 7월 9 03:59 /lib/systemd/system/default.target -> graphical.target

root@server:~# ln -sf /lib/systemd/system/multi-user.target /lib/systemd/system/default.target

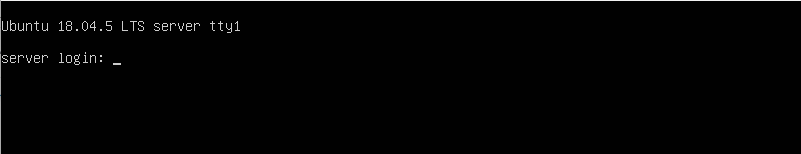
root@server:~# ls -al /lib/systemd/system/default.target

lrwxrwxrwx 1 root root 37 9월 7 11:33 /lib/systemd/system/default.target -> /lib/systemd/system/multi-user.target

### 재실행

root@server:~# reboot

### CLI 형태로 실행되는 것을 확인



### 로그인 후 startx 명령어를 이용해서 X Window 환경을 이용

### 터미널에서 default.target을 변경 후 reboot

ubuntu@server:~$ sudo ln -sf /lib/systemd/system/graphical.target /lib/systemd/system/default.target

[sudo] ubuntu의 암호: ubuntu

ubuntu@server:~$ ls -al /lib/systemd/system/default.target

lrwxrwxrwx 1 root root 36 9월 7 11:38 /lib/systemd/system/default.target -> /lib/systemd/system/graphical.target

ubuntu@server:~$ sudo reboot

### GUI 형태로 실행되는 것을 확인

## 프롬프트 의미

사용자를 구분 ⇒ $: 일반사용자, #: 루트사용자

-

ubuntu@server:~$ ls -al /lib/systemd/system/default.target

------ ------ - ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ 명령어 ⇒ 명령어 [서브명령어] [옵션] [파라미터]

| | |

| | +--- 현재 명령어를 입력하고 있는 위치 (디렉터리)

| | ~ ⇒ 로그인한 계정(사용자)의 홈 디렉터리 ⇒ root → /root, 일반사용자 → /home/계정명

| +--------- 호스트 명

+---------------- 로그인한 계정

## 명령어 사용법을 확인

### 방법1. --help 옵션을 이용

ubuntu@server:~$ ls **--help**

사용법: ls [<옵션>]... [<파일>]...

List information about the FILEs (the current directory by default).

Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all do not ignore entries starting with .

-A, --almost-all do not list implied . and ..

--author with -l, print the author of each file

-b, --escape print C-style escapes for nongraphic characters

--block-size=SIZE scale sizes by SIZE before printing them; e.g.,

'--block-size=M' prints sizes in units of

1,048,576 bytes; see SIZE format below

-B, --ignore-backups do not list implied entries ending with ~

-c with -lt: sort by, and show, ctime (time of last

modification of file status information);

with -l: show ctime and sort by name;

otherwise: sort by ctime, newest first

-C list entries by columns

--color[=WHEN] colorize the output; WHEN can be 'always' (default

if omitted), 'auto', or 'never'; more info below

-d, --directory list directories themselves, not their contents

-D, --dired generate output designed for Emacs' dired mode

-f do not sort, enable -aU, disable -ls --color

-F, --classify append indicator (one of \*/=>@|) to entries

--file-type likewise, except do not append '\*'

--format=WORD across -x, commas -m, horizontal -x, long -l,

single-column -1, verbose -l, vertical -C

--full-time like -l --time-style=full-iso

-g like -l, but do not list owner

--group-directories-first

group directories before files;

can be augmented with a --sort option, but any

use of --sort=none (-U) disables grouping

-G, --no-group in a long listing, don't print group names

-h, --human-readable with -l and/or -s, print human readable sizes

(e.g., 1K 234M 2G)

--si likewise, but use powers of 1000 not 1024

-H, --dereference-command-line

follow symbolic links listed on the command line

--dereference-command-line-symlink-to-dir

follow each command line symbolic link

that points to a directory

--hide=PATTERN do not list implied entries matching shell PATTERN

(overridden by -a or -A)

--hyperlink[=WHEN] hyperlink file names; WHEN can be 'always'

(default if omitted), 'auto', or 'never'

--indicator-style=WORD append indicator with style WORD to entry names:

none (default), slash (-p),

file-type (--file-type), classify (-F)

-i, --inode print the index number of each file

-I, --ignore=PATTERN do not list implied entries matching shell PATTERN

-k, --kibibytes default to 1024-byte blocks for disk usage

-l use a long listing format

-L, --dereference when showing file information for a symbolic

link, show information for the file the link

references rather than for the link itself

-m fill width with a comma separated list of entries

-n, --numeric-uid-gid like -l, but list numeric user and group IDs

-N, --literal print entry names without quoting

-o like -l, but do not list group information

-p, --indicator-style=slash

append / indicator to directories

-q, --hide-control-chars print ? instead of nongraphic characters

--show-control-chars show nongraphic characters as-is (the default,

unless program is 'ls' and output is a terminal)

-Q, --quote-name enclose entry names in double quotes

--quoting-style=WORD use quoting style WORD for entry names:

literal, locale, shell, shell-always,

shell-escape, shell-escape-always, c, escape

-r, --reverse reverse order while sorting

-R, --recursive list subdirectories recursively

-s, --size print the allocated size of each file, in blocks

-S sort by file size, largest first

--sort=WORD sort by WORD instead of name: none (-U), size (-S),

time (-t), version (-v), extension (-X)

--time=WORD with -l, show time as WORD instead of default

modification time: atime or access or use (-u);

ctime or status (-c); also use specified time

as sort key if --sort=time (newest first)

--time-style=STYLE with -l, show times using style STYLE:

full-iso, long-iso, iso, locale, or +FORMAT;

FORMAT is interpreted like in 'date'; if FORMAT

is FORMAT1<newline>FORMAT2, then FORMAT1 applies

to non-recent files and FORMAT2 to recent files;

if STYLE is prefixed with 'posix-', STYLE

takes effect only outside the POSIX locale

-t sort by modification time, newest first

-T, --tabsize=COLS assume tab stops at each COLS instead of 8

-u with -lt: sort by, and show, access time;

with -l: show access time and sort by name;

otherwise: sort by access time, newest first

-U do not sort; list entries in directory order

-v natural sort of (version) numbers within text

-w, --width=COLS set output width to COLS. 0 means no limit

-x list entries by lines instead of by columns

-X sort alphabetically by entry extension

-Z, --context print any security context of each file

-1 list one file per line. Avoid '\n' with -q or -b

--help 이 도움말을 표시하고 끝냅니다

--version 버전 정보를 출력하고 끝냅니다

### 방법2. man 도움말

ubuntu@server:~$ man ls

## 명령어 옵션을 사용하는 방법

### -영문자 또는 --단어 ⇒ -영문자 방식은 여러개의 옵션을 조합해서 사용할 때

ubuntu@server:~$ ls -a

. .profile examples.desktop

.. .ssh test

.ICEauthority .sudo\_as\_admin\_successful 공개

.Xauthority .vboxclient-clipboard.pid 다운로드

.bash\_history .vboxclient-display-svga-x11.pid 문서

.bash\_logout .vboxclient-draganddrop.pid 바탕화면

.bashrc .vboxclient-seamless.pid 비디오

.cache .xsession-errors 사진

.config 2nd.txt 음악

.gnupg 3rd.txt 템플릿

.local 4th.txt

.mozilla abc.txt

ubuntu@server:~$ ls --all

. .profile examples.desktop

.. .ssh test

.ICEauthority .sudo\_as\_admin\_successful 공개

.Xauthority .vboxclient-clipboard.pid 다운로드

.bash\_history .vboxclient-display-svga-x11.pid 문서

.bash\_logout .vboxclient-draganddrop.pid 바탕화면

.bashrc .vboxclient-seamless.pid 비디오

.cache .xsession-errors 사진

.config 2nd.txt 음악

.gnupg 3rd.txt 템플릿

.local 4th.txt

.mozilla abc.txt

### -영문자영문자 형식으로 조합된 경우 ⇒ -영문자 -영문자 옵션의 결합

ubuntu@server:~$ ls -al ⇐ -a 옵션과 -l 옵션을 결합 ⇒ ls -a -l 과 동일

합계 124

drwxr-xr-x 16 ubuntu ubuntu 4096 9월 7 11:51 .

drwxr-xr-x 3 root root 4096 9월 4 10:04 ..

-rw------- 1 ubuntu ubuntu 1908 9월 7 11:39 .ICEauthority

-rw------- 1 ubuntu ubuntu 0 9월 7 11:39 .Xauthority

-rw------- 1 ubuntu ubuntu 927 9월 7 11:49 .bash\_history

-rw-r--r-- 1 ubuntu ubuntu 220 9월 4 10:04 .bash\_logout

-rw-r--r-- 1 ubuntu ubuntu 3771 9월 4 10:04 .bashrc

drwx------ 13 ubuntu ubuntu 4096 9월 7 08:34 .cache

drwx------ 13 ubuntu ubuntu 4096 9월 7 11:47 .config

drwx------ 3 ubuntu ubuntu 4096 9월 7 11:37 .gnupg

drwx------ 3 ubuntu ubuntu 4096 9월 4 13:28 .local

drwx------ 5 ubuntu ubuntu 4096 9월 4 15:39 .mozilla

-rw-r--r-- 1 ubuntu ubuntu 807 9월 4 10:04 .profile

drwx------ 2 ubuntu ubuntu 4096 9월 7 08:36 .ssh

-rw-r--r-- 1 ubuntu ubuntu 0 9월 4 15:10 .sudo\_as\_admin\_successful

-rw-r----- 1 ubuntu ubuntu 5 9월 7 11:39 .vboxclient-clipboard.pid

-rw-r----- 1 ubuntu ubuntu 5 9월 7 11:39 .vboxclient-display-svga-x11.pid

-rw-r----- 1 ubuntu ubuntu 5 9월 7 11:39 .vboxclient-draganddrop.pid

-rw-r----- 1 ubuntu ubuntu 5 9월 7 11:39 .vboxclient-seamless.pid

-rw------- 1 ubuntu ubuntu 4269 9월 7 11:36 .xsession-errors

-rw-r--r-- 1 ubuntu ubuntu 0 9월 4 13:33 2nd.txt

-rw-r--r-- 1 ubuntu ubuntu 0 9월 4 13:58 3rd.txt

-rw-r--r-- 1 ubuntu ubuntu 0 9월 4 13:58 4th.txt

-rw-r--r-- 1 ubuntu ubuntu 0 9월 4 13:29 abc.txt

-rw-r--r-- 1 ubuntu ubuntu 8980 9월 4 10:04 examples.desktop

-rw-r--r-- 1 ubuntu ubuntu 64 9월 7 11:48 test

drwxr-xr-x 2 ubuntu ubuntu 4096 9월 4 13:28 공개

drwxr-xr-x 2 ubuntu ubuntu 4096 9월 4 13:28 다운로드

drwxr-xr-x 2 ubuntu ubuntu 4096 9월 4 13:28 문서

drwxr-xr-x 2 ubuntu ubuntu 4096 9월 4 13:28 바탕화면

drwxr-xr-x 2 ubuntu ubuntu 4096 9월 4 13:28 비디오

drwxr-xr-x 2 ubuntu ubuntu 4096 9월 4 13:28 사진

drwxr-xr-x 2 ubuntu ubuntu 4096 9월 4 13:28 음악

drwxr-xr-x 2 ubuntu ubuntu 4096 9월 4 13:28 템플릿

## Ubuntu Server에서 CD-ROM을 마운트

C:\Program Files\Oracle\VirtualBox\VBoxGuestAdditions.iso

### #1 root 계정으로 로그인

ubuntu@server-b:~$ sudo su ⇐ root 계정으로 로그인

[sudo] password for ubuntu: ubuntu

root@server-b:/home/ubuntu# cd ⇐ cd 명령어는 디렉터리를 변경. 인자가 없는 경우에는 사용자의 홈 디렉터로 이동

root@server-b:~# ⇐ root 사용자의 홈 디렉터리로 이동

root@server-b:~# pwd ⇐ 현재 디렉터리를 표시

/root

### #2 CD-ROM을 마운트할 디렉터리를 생성

root@server-b:~# ls /mnt/cdrom

ls: cannot access '/mnt/cdrom': No such file or directory

root@server-b:~# mkdir -p /mnt/cdrom ⇐ mkdir : 디렉터리를 생성

-p : 디렉터리를 순차적으로 생성

root@server-b:~# ls /mnt/cdrom

### #3 마운트

root@server-b:~# **mount /dev/cdrom /mnt/cdrom** ⇐ /dev/cdrom 디바이스를 /mnt/cdrom 디렉터리에 연결

mount: /mnt/cdrom: WARNING: device write-protected, mounted read-only.

root@server-b:~# cd /mnt/cdrom ⇐ 마운트된 디렉터리로 접근해서 디바이스를 이용

root@server-b:/mnt/cdrom# ls

AUTORUN.INF runasroot.sh VBoxSolarisAdditions.pkg

autorun.sh TRANS.TBL VBoxWindowsAdditions-amd64.exe

cert VBoxDarwinAdditions.pkg VBoxWindowsAdditions.exe

NT3x VBoxDarwinAdditionsUninstall.tool VBoxWindowsAdditions-x86.exe

OS2 VBoxLinuxAdditions.run

root@server-b:/mnt/cdrom# ./VBoxLinuxAdditions.run --nox11 ⇐ VirtualBox 확장 프로그램 실행

Verifying archive integrity... All good.

Uncompressing VirtualBox 6.1.12 Guest Additions for Linux........

VirtualBox Guest Additions installer

Removing installed version 6.1.12 of VirtualBox Guest Additions...

Copying additional installer modules ...

Installing additional modules ...

VirtualBox Guest Additions: Starting.

VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel

modules. This may take a while.

VirtualBox Guest Additions: To build modules for other installed kernels, run

VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup <version>

VirtualBox Guest Additions: or

VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup all

VirtualBox Guest Additions: Building the modules for kernel 4.15.0-115-generic.

This system is currently not set up to build kernel modules.

Please install the gcc make perl packages from your distribution.

VirtualBox Guest Additions: Running kernel modules will not be replaced until

the system is restarted

root@server-b:/mnt/cdrom#

root@server-b:/mnt/cdrom# mount

sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)

:

/var/lib/snapd/snaps/core\_8268.snap on /snap/core/8268 type squashfs (ro,nodev,relatime,x-gdu.hide)

lxcfs on /var/lib/lxcfs type fuse.lxcfs (rw,nosuid,nodev,relatime,user\_id=0,group\_id=0,allow\_other)

tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=204096k,mode=700,uid=1000,gid=1000)

/dev/sr0 on /mnt/cdrom type iso9660 (ro,relatime,nojoliet,check=s,map=n,blocksize=2048)

root@server-b:/mnt/cdrom#

### #4 언마운트

root@server-b:/mnt/cdrom# umount /mnt/cdrom

umount: /mnt/cdrom: target is busy. ⇐ 마운트 디렉터리(/mnt/cdrom)에서 umount하려고 했기 때문에 발생

root@server-b:/mnt/cdrom# umount --help

Usage:

umount [-hV]

umount -a [options]

umount [options] <source> | <directory>

Unmount filesystems.

Options:

-a, --all unmount all filesystems

-A, --all-targets unmount all mountpoints for the given device in the

current namespace

-c, --no-canonicalize don't canonicalize paths

-d, --detach-loop if mounted loop device, also free this loop device

--fake dry run; skip the umount(2) syscall

-f, --force force unmount (in case of an unreachable NFS system)

-i, --internal-only don't call the umount.<type> helpers

-n, --no-mtab don't write to /etc/mtab

-l, --lazy detach the filesystem now, clean up things later

-O, --test-opts <list> limit the set of filesystems (use with -a)

-R, --recursive recursively unmount a target with all its children

-r, --read-only in case unmounting fails, try to remount read-only

-t, --types <list> limit the set of filesystem types

-v, --verbose say what is being done

-h, --help display this help

-V, --version display version

For more details see umount(8).

root@server-b:/mnt/cdrom# umount -l /mnt/cdrom

root@server-b:/mnt/cdrom# mount ⇐ cdrom이 사라진 것을 확인

sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)

:

lxcfs on /var/lib/lxcfs type fuse.lxcfs (rw,nosuid,nodev,relatime,user\_id=0,group\_id=0,allow\_other)

tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=204096k,mode=700,uid=1000,gid=1000)

root@server-b:/mnt/cdrom#

## 가상 머신의 특정 디렉터리를 ISO 파일로 제작 (P135)

### #1 genisoimage 프로그램 설치 여부를 확인

root@server-b:/# dpkg --get-selections genisoimage

dpkg: no packages found matching genisoimage

### #2 genisoimage 설치

root@server-b:/# apt install genisoimage

### #3 /bin 디렉터리 아래에 있는 파일과 디렉터리를 boot.iso 파일로 압축

root@server-b:/# genisoimage -r -J -o boot.iso /boot

I: -input-charset not specified, using utf-8 (detected in locale settings)

Using MENU000.LST;1 for /boot/grub/menu.lst (menu.lst~)

Using GCRY\_000.MOD;1 for /boot/grub/i386-pc/gcry\_sha512.mod (gcry\_sha1.mod)

Using PASSW000.MOD;1 for /boot/grub/i386-pc/password\_pbkdf2.mod (password.mod)

Using MDRAI000.MOD;1 for /boot/grub/i386-pc/mdraid09\_be.mod (mdraid09.mod)

Using SEARC000.MOD;1 for /boot/grub/i386-pc/search\_fs\_file.mod (search\_fs\_uuid.mod)

Using GFXTE000.MOD;1 for /boot/grub/i386-pc/gfxterm\_menu.mod (gfxterm\_background.mod)

Using GCRY\_001.MOD;1 for /boot/grub/i386-pc/gcry\_sha1.mod (gcry\_sha256.mod)

Using USBSE000.MOD;1 for /boot/grub/i386-pc/usbserial\_usbdebug.mod (usbserial\_ftdi.mod)

Using VIDEO000.MOD;1 for /boot/grub/i386-pc/videotest.mod (videotest\_checksum.mod)

Using USBSE001.MOD;1 for /boot/grub/i386-pc/usbserial\_ftdi.mod (usbserial\_pl2303.mod)

Using USBSE002.MOD;1 for /boot/grub/i386-pc/usbserial\_pl2303.mod (usbserial\_common.mod)

Using PART\_000.MOD;1 for /boot/grub/i386-pc/part\_sunpc.mod (part\_sun.mod)

Using XNU\_U000.MOD;1 for /boot/grub/i386-pc/xnu\_uuid.mod (xnu\_uuid\_test.mod)

Using MULTI000.MOD;1 for /boot/grub/i386-pc/multiboot2.mod (multiboot.mod)

13.12% done, estimate finish Mon Sep 7 05:34:28 2020

26.25% done, estimate finish Mon Sep 7 05:34:28 2020

39.33% done, estimate finish Mon Sep 7 05:34:28 2020

52.46% done, estimate finish Mon Sep 7 05:34:28 2020

65.55% done, estimate finish Mon Sep 7 05:34:28 2020

78.67% done, estimate finish Mon Sep 7 05:34:28 2020

91.76% done, estimate finish Mon Sep 7 05:34:28 2020

Total translation table size: 0

Total rockridge attributes bytes: 25187

Total directory bytes: 45056

Path table size(bytes): 66

Max brk space used 44000

38143 extents written (74 MB)

### #4 ISO 파일을 마운트

### #4-1 마운트 디렉터리 생성

root@server-b:/# mkdir -p /mnt/iso

### #4-2 mount

root@server-b:/# **mount -o loop boot.iso** /mnt/iso

mount: /mnt/iso: WARNING: device write-protected, mounted read-only.

### #5 /bin 디렉터리와 /mnt/iso 디렉터리를 비교

root@server-b:/# ls -l /mnt/iso

total 68985

-r--r--r-- 1 root root 217469 Aug 26 13:45 config-4.15.0-115-generic

dr-xr-xr-x 5 root root 2048 Sep 6 23:49 grub

-r--r--r-- 1 root root 57957373 Sep 6 23:52 initrd.img-4.15.0-115-generic

-r--r--r-- 1 root root 4074919 Aug 26 13:45 System.map-4.15.0-115-generic

-r--r--r-- 1 root root 8388256 Aug 26 13:48 vmlinuz-4.15.0-115-generic

root@server-b:/# ls -l /boot

total 68992

-rw-r--r-- 1 root root 217469 Aug 26 13:45 config-4.15.0-115-generic

drwxr-xr-x 5 root root 4096 Sep 6 23:49 grub

-rw-r--r-- 1 root root 57957373 Sep 6 23:52 initrd.img-4.15.0-115-generic

-rw------- 1 root root 4074919 Aug 26 13:45 System.map-4.15.0-115-generic

-rw------- 1 root root 8388256 Aug 26 13:48 vmlinuz-4.15.0-115-generic

### #6 umount

root@server-b:/# umount /mnt/iso

## 기본 명령어

root@server-b:/# cd ⇐ root 사용자의 홈 디렉터리로 이동

root@server-b:~# ls ⇐ 서브 디렉터리와 파일을 보여 달라

root@server-b:~# ls -a ⇐ 숨김 파일을 포함해서 출력

. .. .bash\_history .bashrc .lesshst .profile .ssh ⇐ 마침표로 시작하면 → 숨김 파일 및 디렉터리

Quiz. 명령어의 실행 결과가 나머지와 다른 것은?

1. root@server:/bin# ls
2. root@server:/bin# ls . ⇐ . 또는 .. 으로 시작하는 경로 ⇒ 상대 경로
3. root@server:/bin# ls ./
4. root@server:/bin# ls / ⇐ / 로 시작하는 경로 ⇒ 절대 경로
5. root@server:/bin# ls /bin
6. root@server:/bin# ls /bin/\*
7. root@server:/bin# ls /bin/

Quiz. root 사용자의 홈 디렉터로 이동하는 방법이 아닌 것은? (답 없음)

1. root@server:/bin/test# cd
2. root@server:/bin/test# cd ~
3. root@server:/bin/test# cd $HOME
4. root@server:/bin/test# cd /root

Quiz. 다음 명령어의 실행 결과는?

root@server:/tmp# touch aaa ⇐ /tmp/aaa 파일 생성

root@server:/tmp# touch bbb ⇐ /tmp/bbb 파일 생성

root@server:/tmp# touch ccc ⇐ /tmp/ccc 파일 생성

root@server:/tmp# mkdir ddd ⇐ /tmp/ddd/ 디렉터리 생성

root@server:/tmp# ls ⇐ /tmp 아래의 파일과 디렉터리를 출력 → aaa bbb ccc ddd

root@server:/tmp# mv aaa bbb ccc ddd ⇐ mv 이전할대상들 이전할위치(디렉터리)

root@server:/tmp# ls ⇐ ddd 디렉터리만 출력

root@server:/tmp# ls ./ddd ⇐ /tmp/ddd 디렉터리 아래의 파일과 디렉터리를 출력 → aaa bbb ccc

Quiz. 다음 명령어의 실행 결과는?

root@server:/tmp# touch aaa

root@server:/tmp# touch bbb

root@server:/tmp# touch ccc

root@server:/tmp# mkdir ddd

root@server:/tmp# mkdir eee

root@server:/tmp# ls

root@server:/tmp# mv aaa bbb ccc ddd eee

root@server:/tmp# ls

root@server:/tmp# ls ./ddd ⇐ mv 명령어로 /tmp/eee 디렉터리 아래오 이동했기 때문에

No such file or dirctory 오류가 발생

root@server:/tmp# ls ./eee → aaa bbb ccc ddd ⇐ 3개의 파일과 1개의 디렉터리를 확인

15시 40분에 이어서 진행하겠습니다.

## 네트워크 정보 확인

root@server-b:~# ifconfig ⇐ IP 정보를 확인

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 10.0.2.5 netmask 255.255.255.0 broadcast 10.0.2.255 ⇐ IP 주소와 서브넷 마스크를 확인

inet6 fe80::a00:27ff:fed9:ce01 prefixlen 64 scopeid 0x20<link>

ether 08:00:27:d9:ce:01 txqueuelen 1000 (Ethernet)

RX packets 50435 bytes 70241350 (70.2 MB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 8091 bytes 741383 (741.3 KB)

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 1000 (Local Loopback)

RX packets 100 bytes 12950 (12.9 KB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 100 bytes 12950 (12.9 KB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@server-b:~# ifconfig enp0s3

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 10.0.2.5 netmask 255.255.255.0 broadcast 10.0.2.255

inet6 fe80::a00:27ff:fed9:ce01 prefixlen 64 scopeid 0x20<link>

ether 08:00:27:d9:ce:01 txqueuelen 1000 (Ethernet)

RX packets 50487 bytes 70245070 (70.2 MB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 8143 bytes 747079 (747.0 KB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@server-b:~# ip route ⇐ 게이트웨이 주소를 확인

default via 10.0.2.1 dev enp0s3 proto dhcp src 10.0.2.5 metric 100

10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.5

10.0.2.1 dev enp0s3 proto dhcp scope link src 10.0.2.5 metric 100

root@server-b:~# systemd-resolve --status enp0s3 ⇐ DNS 정보를 확인

Link 2 (enp0s3)

Current Scopes: DNS

LLMNR setting: yes

MulticastDNS setting: no

DNSSEC setting: no

DNSSEC supported: no

DNS Servers: 168.126.63.1

168.126.63.2

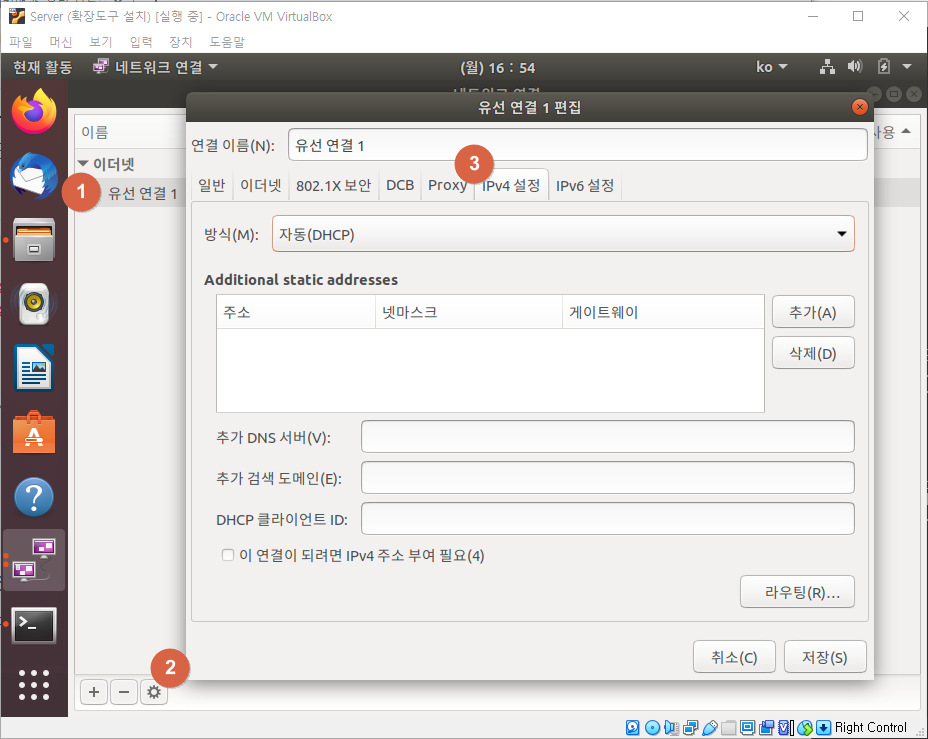
16시 50분에 이어서 실습을 진행하겠습니다.

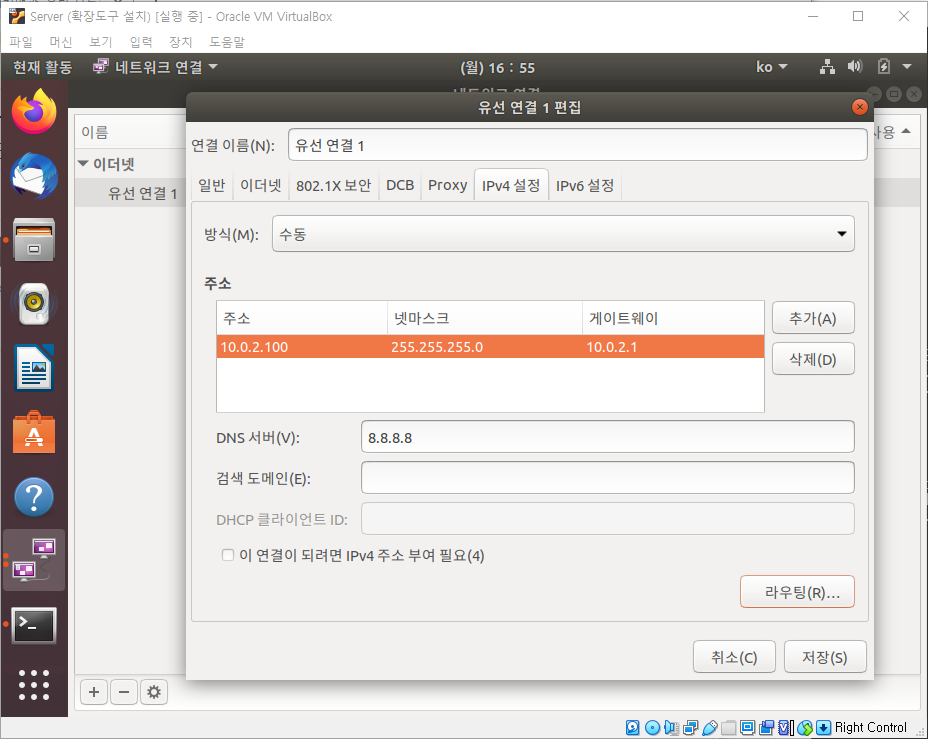
## 고정 IP 주소로 변경 (P153 참고)

### Ubuntu Desktop에서 고정 IP 설정

root@server:~# nm-connection-editor

IPv4 설정





* 방식: 수동 ⇒ DHCP를 사용하지 않음
* 주소: 10.0.2.100 ⇒ VirtualBox의 NatNetwork에서 부여할 수 있는 IP 범위내의 값을 사용
* 넷마스크: 255.255.255.0
* 게이트웨이: 10.0.2.1
* DNS 서버: 8.8.8.8 ⇒ 구글에서 제공하는 DNS 서버 주소

root@server:~# reboot

ubuntu@server:~$ ip a

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000

link/ether 08:00:27:96:b2:75 brd ff:ff:ff:ff:ff:ff

inet 10.0.2.100/24 brd 10.0.2.255 scope global noprefixroute enp0s3

valid\_lft forever preferred\_lft forever

inet6 fe80::2e79:698a:dba3:9968/64 scope link noprefixroute

valid\_lft forever preferred\_lft forever

### Ubuntu Server에서 고정 IP를 설정

ubuntu@server-b:~$ sudo su

[sudo] password for ubuntu: ubuntu

root@server-b:/home/ubuntu# cd

root@server-b:~# ls /etc/netplan/ ⇐ 네트워크 설정 파일을 확인

50-cloud-init.yaml

root@server-b:~# cp /etc/netplan/50-cloud-init.yaml /etc/netplan/50-cloud-init.yaml.bak ⇐ 백업 파일 생성

root@server-b:~# vi /etc/netplan/50-cloud-init.yaml ⇐ 네트워크 설정 파일을 편집

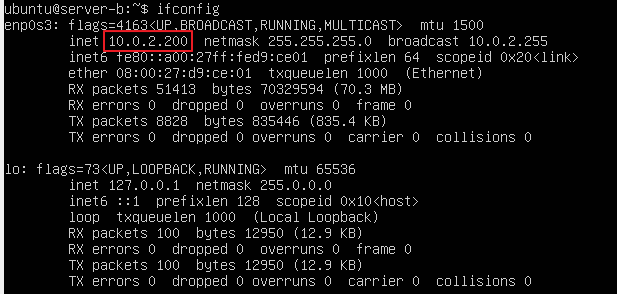
|  |
| --- |
| network:  ethernets:  enp0s3:  dhcp4: false  addresses: [10.0.2.200/24]  gateway4: 10.0.2.1  nameservers:  addresses: [8.8.8.8]  version: 2 |

\* 주의할 점

탭(tab)을 사용하면 안 됨 (반드시 공백을 사용)

같은 레벨의 정보는 동일 컬럼에 맞춰서 작성

root@server-b:~# netplan apply



### 각각 접속 확인

Ubuntu Desktop에서 Ubuntu Server로 접속 ⇒ ssh 10.0.2.200

Ubuntu Server에서 Ubuntu Desktop으로 접속 ⇒ ssh 10.0.2.100



호스트 PC에서 Ubuntu Server로 접속 ⇒ C:\Users\myanj> ssh ubuntu@192.168.0.218 -p 22

호스트 PC에서 Ubuntu Desktop으로 접속 ⇒ C:\Users\myanj> ssh ubuntu@192.168.0.218 -p 222

9/8 (화)

https://docs.google.com/presentation/d/1uK7u-XNcx6SSby0YqE9VQV4lBDNeO3w89I2JajetoWQ/edit?usp=sharing

https://docs.google.com/presentation/d/1dso8TN5Tb2CfT7fQ8zZ0fauCidZobbNwvZNRUweK708/edit?usp=sharing

https://docs.google.com/presentation/d/1GPoOjxa7Eg\_S9\_r9FL64l\_a1QK7kjG3iNl139yjmMKs/edit?usp=sharing

Quiz.

ubuntu@server:~$ date > now1

ubuntu@server:~$ date > now2

ubuntu@server:~$ cat now1

2020. 09. 08. (화) 09:20:37 KST

ubuntu@server:~$ cat now2

2020. 09. 08. (화) 09:20:43 KST

ubuntu@server:~$ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

아래 명령어의 실행 결과가 다음과 같이 나올 수 있도록 \_\_\_\_ 부분에 명령어를 완성하세요.

ubuntu@server:~$ cat now3

2020. 09. 08. (화) 09:20:37 KST

2020. 09. 08. (화) 09:20:43 KST

(정답예) $ cat now1 now2 > now3

## passwd 파일, group 파일

ubuntu@server:~$ sudo su -

[sudo] ubuntu의 암호: ubuntu

root@server:~# cat /etc/passwd ⇐ 사용자 계정 정보가 저장

root:x:0:0:root:/root:/bin/bash ⇐ root 계정

daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

bin:x:2:2:bin:/bin:/usr/sbin/nologin

sys:x:3:3:sys:/dev:/usr/sbin/nologin

sync:x:4:65534:sync:/bin:/bin/sync

games:x:5:60:games:/usr/games:/usr/sbin/nologin

man:x:6:12:man:/var/cache/man:/usr/sbin/nologin

lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin

mail:x:8:8:mail:/var/mail:/usr/sbin/nologin

news:x:9:9:news:/var/spool/news:/usr/sbin/nologin

uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin

proxy:x:13:13:proxy:/bin:/usr/sbin/nologin

www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin

backup:x:34:34:backup:/var/backups:/usr/sbin/nologin

list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin

irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin

gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin

nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin

systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin

systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin

syslog:x:102:106::/home/syslog:/usr/sbin/nologin

messagebus:x:103:107::/nonexistent:/usr/sbin/nologin

\_apt:x:104:65534::/nonexistent:/usr/sbin/nologin

uuidd:x:105:111::/run/uuidd:/usr/sbin/nologin

avahi-autoipd:x:106:112:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin

usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin

dnsmasq:x:108:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin

rtkit:x:109:114:RealtimeKit,,,:/proc:/usr/sbin/nologin

cups-pk-helper:x:110:116:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin

speech-dispatcher:x:111:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false

whoopsie:x:112:117::/nonexistent:/bin/false

kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin

saned:x:114:119::/var/lib/saned:/usr/sbin/nologin

pulse:x:115:120:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin

avahi:x:116:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin

colord:x:117:123:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin

hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false

geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin

gnome-initial-setup:x:120:65534::/run/gnome-initial-setup/:/bin/false

gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false

ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash ⇐ ubuntu 계정

vboxadd:x:999:1::/var/run/vboxadd:/bin/false

sshd:x:122:65534::/run/sshd:/usr/sbin/nologin

사용자이름:비밀번호:사용자 ID:사용자 소속 그룹 ID:추가 정보:홈 디렉터리: 기본 셸

~~~~~~~~

x ⇒ shadow 파일로 이전

root@server:~# cat /etc/group

root:x:0:

daemon:x:1:

bin:x:2:

sys:x:3:

adm:x:4:syslog,ubuntu

tty:x:5:

disk:x:6:

lp:x:7:

mail:x:8:

news:x:9:

uucp:x:10:

man:x:12:

proxy:x:13:

kmem:x:15:

dialout:x:20:

fax:x:21:

voice:x:22:

cdrom:x:24:ubuntu

floppy:x:25:

tape:x:26:

sudo:x:27:ubuntu

audio:x:29:pulse

dip:x:30:ubuntu

www-data:x:33:

backup:x:34:

operator:x:37:

list:x:38:

irc:x:39:

src:x:40:

gnats:x:41:

shadow:x:42:

utmp:x:43:

video:x:44:

sasl:x:45:

plugdev:x:46:ubuntu

staff:x:50:

games:x:60:

users:x:100:

nogroup:x:65534:

systemd-journal:x:101:

systemd-network:x:102:

systemd-resolve:x:103:

input:x:104:

crontab:x:105:

syslog:x:106:

messagebus:x:107:

netdev:x:108:

mlocate:x:109:

ssl-cert:x:110:

uuidd:x:111:

avahi-autoipd:x:112:

bluetooth:x:113:

rtkit:x:114:

ssh:x:115:

lpadmin:x:116:ubuntu

whoopsie:x:117:

scanner:x:118:saned

saned:x:119:

pulse:x:120:

pulse-access:x:121:

avahi:x:122:

colord:x:123:

geoclue:x:124:

gdm:x:125:

ubuntu:x:1000:

sambashare:x:126:ubuntu

vboxsf:x:999:

그룹 이름:비밀번호:그룹 ID:보조 그룹 사용자

## adduser

root@server:~# adduser --help

adduser [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]

[--firstuid ID] [--lastuid ID] [--gecos GECOS] [--ingroup GROUP | --gid ID]

[--disabled-password] [--disabled-login] [--add\_extra\_groups]

[--encrypt-home] USER

Add a normal user

adduser --system [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]

[--gecos GECOS] [--group | --ingroup GROUP | --gid ID] [--disabled-password]

[--disabled-login] [--add\_extra\_groups] USER

Add a system user

adduser --group [--gid ID] GROUP

addgroup [--gid ID] GROUP

Add a user group

addgroup --system [--gid ID] GROUP

Add a system group

adduser USER GROUP

Add an existing user to an existing group

general options:

--quiet | -q don't give process information to stdout

--force-badname allow usernames which do not match the

NAME\_REGEX[\_SYSTEM] configuration variable

--extrausers uses extra users as the database

--help | -h usage message

--version | -v version number and copyright

--conf | -c FILE use FILE as configuration file

## usermod

root@server:~# usermod --help

Usage: usermod [options] LOGIN

Options:

-c, --comment COMMENT new value of the GECOS field

-d, --home HOME\_DIR new home directory for the user account

-e, --expiredate EXPIRE\_DATE set account expiration date to EXPIRE\_DATE

-f, --inactive INACTIVE set password inactive after expiration

to INACTIVE

-g, --gid GROUP force use GROUP as new primary group

-G, --groups GROUPS new list of supplementary GROUPS

-a, --append append the user to the supplemental GROUPS

mentioned by the -G option without removing

him/her from other groups

-h, --help display this help message and exit

-l, --login NEW\_LOGIN new value of the login name

-L, --lock lock the user account

-m, --move-home move contents of the home directory to the

new location (use only with -d)

-o, --non-unique allow using duplicate (non-unique) UID

-p, --password PASSWORD use encrypted password for the new password

-R, --root CHROOT\_DIR directory to chroot into

-s, --shell SHELL new login shell for the user account

-u, --uid UID new UID for the user account

-U, --unlock unlock the user account

-v, --add-subuids FIRST-LAST add range of subordinate uids

-V, --del-subuids FIRST-LAST remove range of subordinate uids

-w, --add-subgids FIRST-LAST add range of subordinate gids

-W, --del-subgids FIRST-LAST remove range of subordinate gids

-Z, --selinux-user SEUSER new SELinux user mapping for the user account

잠시 쉬고, 10시 15분에 이어서 진행하겠습니다.

# 실습5-1 사용자와 그룹관리 실습 (P179)

### #1 hanbit1 사용자 계정 추가

root@server:~# adduser hanbit1

'hanbit1' 사용자를 추가 중...

새 그룹 'hanbit1' (1001) 추가 ...

새 사용자 'hanbit1' (1001) 을(를) 그룹 'hanbit1' (으)로 추가 ...

'/home/hanbit1' 홈 디렉터리를 생성하는 중...

'/etc/skel'에서 파일들을 복사하는 중...

새 UNIX 암호 입력: 1234

새 UNIX 암호 재입력: 1234

passwd: 암호를 성공적으로 업데이트했습니다

hanbit1의 사용자의 정보를 바꿉니다

새로운 값을 넣거나, 기본값을 원하시면 엔터를 치세요

이름 []:

방 번호 []:

직장 전화번호 []:

집 전화번호 []:

기타 []:

정보가 올바릅니까? [Y/n]

### #2 passwd 파일에 hanbit1 계정 추가를 확인

root@server:~# tail /etc/passwd

avahi:x:116:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin

colord:x:117:123:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin

hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false

geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin

gnome-initial-setup:x:120:65534::/run/gnome-initial-setup/:/bin/false

gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false

ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash

vboxadd:x:999:1::/var/run/vboxadd:/bin/false

sshd:x:122:65534::/run/sshd:/usr/sbin/nologin

hanbit1:x:1001:1001:,,,:/home/hanbit1:/bin/bash

### #3 group 파일에 hanbit1 그룹 추가를 확인

root@server:~# tail /etc/group

pulse:x:120:

pulse-access:x:121:

avahi:x:122:

colord:x:123:

geoclue:x:124:

gdm:x:125:

ubuntu:x:1000:

sambashare:x:126:ubuntu

vboxsf:x:999:

hanbit1:x:1001: ⇐ 별도로 그룹을 지정하지 않으면 자동으로 사용자 이름과 동일한 그룹이 생성

### #4 hanbit1 계정을 삭제

root@server:~# userdel -r hanbit1 ⇐ 사용자 홈 디렉터리를 포함해서 삭제

userdel: hanbit1 mail spool (/var/mail/hanbit1) not found

### #5 ubuntuGroup 그룹을 생성 후 확인

root@server:~# groupadd ubuntuGroup

root@server:~# tail -3 /etc/group

sambashare:x:126:ubuntu

vboxsf:x:999:

ubuntuGroup:x:1001:

### #6 ubuntuGroup 그룹에 속하는 hanbit1, hanbit2 사용자 계정을 생성

root@server:~# adduser --gid 1001 hanbit1

'hanbit1' 사용자를 추가 중...

새 사용자 'hanbit1' (1001) 을(를) 그룹 'ubuntuGroup' (으)로 추가 ...

'/home/hanbit1' 홈 디렉터리를 생성하는 중...

'/etc/skel'에서 파일들을 복사하는 중...

새 UNIX 암호 입력:

새 UNIX 암호 재입력:

passwd: 암호를 성공적으로 업데이트했습니다

hanbit1의 사용자의 정보를 바꿉니다

새로운 값을 넣거나, 기본값을 원하시면 엔터를 치세요

이름 []:

방 번호 []:

직장 전화번호 []:

집 전화번호 []:

기타 []:

정보가 올바릅니까? [Y/n]

root@server:~# adduser --gid 1001 hanbit2

'hanbit2' 사용자를 추가 중...

새 사용자 'hanbit2' (1002) 을(를) 그룹 'ubuntuGroup' (으)로 추가 ...

'/home/hanbit2' 홈 디렉터리를 생성하는 중...

'/etc/skel'에서 파일들을 복사하는 중...

새 UNIX 암호 입력:

새 UNIX 암호 재입력:

passwd: 암호를 성공적으로 업데이트했습니다

hanbit2의 사용자의 정보를 바꿉니다

새로운 값을 넣거나, 기본값을 원하시면 엔터를 치세요

이름 []:

방 번호 []:

직장 전화번호 []:

집 전화번호 []:

기타 []:

정보가 올바릅니까? [Y/n]

root@server:~# tail -5 /etc/passwd

ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash

vboxadd:x:999:1::/var/run/vboxadd:/bin/false

sshd:x:122:65534::/run/sshd:/usr/sbin/nologin

hanbit1:x:1001:1001:,,,:/home/hanbit1:/bin/bash

hanbit2:x:1002:1001:,,,:/home/hanbit2:/bin/bash

### #7 사용자 홈 디렉터리를 확인

root@server:~# ls -l /home/hanbit1

합계 12

-rw-r--r-- 1 hanbit1 ubuntuGroup 8980 9월 8 10:30 examples.desktop <--+

root@server:~# ls -l /home/hanbit2 |

합계 12 |

-rw-r--r-- 1 hanbit2 ubuntuGroup 8980 9월 8 10:30 examples.desktop <--+

|

root@server:~# ls -l /etc/skel |

합계 12 | 사용자가 생성될 때마다

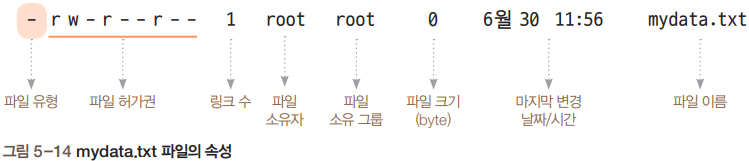
-rw-r--r-- 1 root root 8980 4월 16 2018 examples.desktop -------------+ 사용자 홈 디렉터리로 복제

## 파일 소유권과 허가권 정보

root@server:~# touch mydata.txt

root@server:~# ls -l mydata.txt

**-rw-r--r-- 1 root root** 0 9월 8 10:39 mydata.txt



파일유형

* d ⇒ 디렉터리
* - ⇒ 일반 파일
* b ⇒ 블럭 디바이스(하드디스크, USB, CD/DVD 등)
* c ⇒ 문자 디바이스(키보드, 마우스, 프린터 등)
* l ⇒ 링크



r : 읽기 가능

w : 쓰기 가능

x : 실행 가능

chmod 777 mydata.txt

~~~

⇒ 2진수 → 111 111 111 = rwx rwx rwx ⇒ 소유자, 소유자 그룹, 그 외 모두가 읽고, 쓰고, 실행할 수 있음

## symbolic method 방식

chmod "사용자유형" " + or - " "권한" 파일

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Function** | **Description** |
| u | Who | User (owner) |
| g | Who | Group |
| o | Who | Others |
| a | Who | All |
| = | Operator | Assign |
| + | Operator | Add |
| - | Operator | Remove |
| r | Permission | Read |
| w | Permission | Write |
| x | Permission | Execute |

chmod u+x mydata.txt ⇒ 소유자에 대해서 실행 권한을 부여

chmod a+rw mydata.txt ⇒ 모든 사용자(소유자, 그룹, 그 외)에 대해 읽고, 쓰는 권한을 부여

잠시 쉬고 11시 15분에 이어서 진행하겠습니다.

# 실습 5-2 파일 허가권과 소유권 실행 (P187)

### #1 파일 생성 후 파일 허가권 및 소유권 확인

root@server:~# vi sample

|  |
| --- |
| 쿡북 리눅스를 학습 중입니다.  ls /var |

root@server:~# ls -l sample

-rw-r--r-- 1 root root 49 9월 8 11:28 sample ⇒ 소유자(root)는 읽고 쓸 수 있고, 나머지는 읽기만 가능

### #2 현재 사용자를 확인하고 sample 파일을 실행

root@server:~# whoami

root ⇐ 현재 사용자는 root 인 것을 확인

root@server:~# ./sample

-su: ./sample: 허가 거부 ⇐ 실행 권한이 없기 때문에 오류가 발생

### #3 파일에 모든 사용자가 실행할 수 있도록 실행 권한을 부여 후 실행

rw-r--r-- ⇒ rwx r-x r-x

111 101 101 ⇒ 7 5 5

root@server:~# chmod 755 sample ⇐ 실행 권한 부여

root@server:~# ls -l sample ⇐ 실행 권한이 부여된 것을 확인

-rwxr-xr-x 1 root root 49 9월 8 11:28 sample

root@server:~# ./sample ⇐ 라인 단위로 읽어서 쉘에서 실행하고 결과를 출력

./sample: 줄 1: 쿡북: 명령어를 찾을 수 없음

backups cache crash lib local lock log mail metrics opt run snap spool tmp www

### #4 파일의 소유자를 ubuntu 사용자로 변경

root@server:~# chown ubuntu sample

root@server:~# ls -l sample

-rwxr-xr-x 1 ubuntu root 49 9월 8 11:28 sample ⇐ 파일의 소유자가 ubuntu로 변경

root@server:~# chown ubuntu.ubuntu sample

root@server:~# ls -l sample

-rwxr-xr-x 1 ubuntu ubuntu 49 9월 8 11:28 sample ⇐ 파일의 소유자와 소유자 그룹이 ubuntu로 변경

### #5 ubuntu 사용자로 로그인해서 파일에 접근

root@server:~# su ubuntu ⇐ ubuntu 사용자로 로그인

ubuntu@server:/root$ pwd ⇐ 현재 작업 디렉터리 확인

/root

ubuntu@server:/root$ ls -l /root/sample ⇐ 파일 조회

ls: '/root/sample'에 접근할 수 없습니다: 허가 거부

ubuntu@server:/root$ ls -ld /root

drwx------ 6 root root 4096 9월 8 11:28 /root ⇐ /root 디렉터리는 root 사용자만 읽고 쓰고 실행할 수 있기 때문에

ubuntu 사용자는 /root/sample 파일에 접근할 수 없음

### #6 ubuntu 사용자가 파일을 사용할 수 있도록 홈 디렉터리로 이전 후 실행

ubuntu@server:/root$ exit ⇐ root 사용자로 복귀

exit

root@server:~# mv ./sample ~ubuntu ⇐ 파일을 ubuntu 사용자의 홈 디렉터리로 이전

root@server:~# su ubuntu ⇐ ubuntu 사용자로 로그인

ubuntu@server:/root$ cd ⇐ ubuntu 사용자 홈 디렉터리로 이동

ubuntu@server:~$ ls -l sample ⇐ 파일 조회 (실행 권한 확인)

-rwxr-xr-x 1 ubuntu ubuntu 49 9월 8 11:28 sample

ubuntu@server:~$ ./sample ⇐ 실행

./sample: 줄 1: 쿡북: 명령어를 찾을 수 없음

backups cache crash lib local lock log mail metrics opt run snap spool tmp www

### #7 다시 소유자와 소유자 그룹을 root로 변경

ubuntu@server:~$ ls -l sample ⇐ 파일 소유자, 소유자 그룹 확인

-rwxr-xr-x 1 ubuntu ubuntu 49 9월 8 11:28 sample

ubuntu@server:~$ chown root.root sample ⇐ 파일 소유자, 소유자 그룹을 root로 변경

chown: 'sample'의 소유자 변경: 명령을 허용하지 않음 → 상위 권한의 사용자로 변경할 수 없음

점심 식사하고 13시 10분에 이어서 진행하겠습니다.

# 실습 5-3 하드 링크와 심볼릭 링크 생성

### #1 root 로그인 후 작업 디렉터리를 만들고 파일을 생성

ubuntu@server:~$ sudo su

[sudo] ubuntu의 암호: ubuntu

root@server:/home/ubuntu# cd

root@server:~# mkdir linkdir

root@server:~# cd linkdir

root@server:~/linkdir# echo "원본파일" > orginalfile

root@server:~/linkdir# ls

orginalfile

### #2 하드 링크 파일과 소프트 링크 파일을 생성하고 내용을 확인

root@server:~/linkdir# ln orginalfile hardlink

root@server:~/linkdir# ln -s orginalfile softlink

root@server:~/linkdir# ls -il ⇐ inode 정보를 함께 출력

합계 8

394643 -rw-r--r-- 2 root root 13 9월 8 13:21 hardlink ⇐ 하드 링크는 원본 파일의 inode를 공유

394643 -rw-r--r-- 2 root root 13 9월 8 13:21 orginalfile

394819 lrwxrwxrwx 1 root root 11 9월 8 13:24 softlink -> orginalfile ⇐ 소프트 링크는 별도의 inode를 만들고

root@server:~/linkdir# cat hardlink 원본 파일을 참조

원본파일

root@server:~/linkdir# cat softlink

원본파일

### #3 원본 파일의 경로를 변경한 후 하드 링크 파일과 소프트 링크 파일을 확인

root@server:~/linkdir# mv ./orginalfile ../ ⇐ 원본 파일을 상위 디렉터리로 이전

root@server:~/linkdir# ls -il

합계 4

394643 -rw-r--r-- 2 root root 13 9월 8 13:21 hardlink

394819 lrwxrwxrwx 1 root root 11 9월 8 13:24 softlink -> orginalfile ⇐ orginalfile이 사라져서(링크가 끊겨서)

root@server:~/linkdir# cat hardlink 붉은색으로 표시

원본파일

root@server:~/linkdir# cat softlink

cat: softlink: 그런 파일이나 디렉터리가 없습니다

### #4 원본 파일을 원래 위치로 복원하면 소프트 링크도 복원되는 것을 확인

root@server:~/linkdir# mv ../orginalfile ./

root@server:~/linkdir# ls -il

합계 8

394643 -rw-r--r-- 2 root root 13 9월 8 13:21 hardlink

394643 -rw-r--r-- 2 root root 13 9월 8 13:21 orginalfile

394819 lrwxrwxrwx 1 root root 11 9월 8 13:24 softlink -> orginalfile

root@server:~/linkdir# cat hardlink

원본파일

root@server:~/linkdir# cat orginalfile

원본파일

### #5 원본 파일의 내용을 변경하면 링크 파일도 변경된 내용이 보이는 것을 확인

root@server:~/linkdir# echo "당근이쥐" >> orginalfile

root@server:~/linkdir# cat hardlink

원본파일

당근이쥐

root@server:~/linkdir# cat softlink

원본파일

당근이쥐

### #6 원본 파일을 다른 디렉터리로 옮긴 후 내용을 변경하고 다시 원래 위치로 옮겼을 때 하드 링크와 소프트 링크는 어떻게 내용이 보여질까?

root@server:~/linkdir# mv ./orginalfile ../

root@server:~/linkdir# echo "또 당근이쥐" >> ../orginalfile

root@server:~/linkdir# cat ../orginalfile

원본파일

당근이쥐

또 당근이쥐

root@server:~/linkdir# cat hardlink

원본파일

당근이쥐

또 당근이쥐

root@server:~/linkdir# cat softlink

cat: softlink: 그런 파일이나 디렉터리가 없습니다

root@server:~/linkdir# mv ../orginalfile ./

root@server:~/linkdir# cat orginalfile

원본파일

당근이쥐

또 당근이쥐

root@server:~/linkdir# cat hardlink

원본파일

당근이쥐

또 당근이쥐

root@server:~/linkdir# cat softlink

원본파일

당근이쥐

또 당근이쥐

### #7 원본 파일과 동일한 이름의 파일을 생성하는 경우 하드링크와 소프트링크는 어떻게 동작하는가?

root@server:~/linkdir# mv ./orginalfile ../

root@server:~/linkdir# echo "새로운 원본파일" > orginalfile ⇐ 소프트링크가 만들어진 위치에

root@server:~/linkdir# cat hardlink 원본 파일과 동일한 이름의 파일을 생성

원본파일 ⇐ inode를 공유하고 있기 때문에 앞에서 생성한 원본의 내용을 출력

당근이쥐

또 당근이쥐

root@server:~/linkdir# cat softlink ⇐ 동일한 파일을 찾아서 참조하기 때문에 새롭게 생성된 파일의 내용을 출력

새로운 원본파일

root@server:~/linkdir# ps -ef | grep firefox

ubuntu 3443 1 1 13:50 tty2 00:00:05 /usr/lib/firefox/firefox -new-window

ubuntu 3528 3443 0 13:50 tty2 00:00:00 /usr/lib/firefox/firefox -contentproc -childID 1 -isForBrowser -prefsLen 1 -prefMapSize 224116 -parentBuildID 20200831163820 -appdir /usr/lib/firefox/browser 3443 true tab

ubuntu 3565 3443 0 13:50 tty2 00:00:01 /usr/lib/firefox/firefox -contentproc -childID 2 -isForBrowser -prefsLen 6969 -prefMapSize 224116 -parentBuildID 20200831163820 -appdir /usr/lib/firefox/browser 3443 true tab

ubuntu 3582 3443 0 13:50 tty2 00:00:00 /usr/lib/firefox/firefox -contentproc -childID 3 -isForBrowser -prefsLen 6969 -prefMapSize 224116 -parentBuildID 20200831163820 -appdir /usr/lib/firefox/browser 3443 true tab

ubuntu 3627 3443 0 13:50 tty2 00:00:00 /usr/lib/firefox/firefox -contentproc -childID 4 -isForBrowser -prefsLen 8984 -prefMapSize 224116 -parentBuildID 20200831163820 -appdir /usr/lib/firefox/browser 3443 true tab

root 3699 3388 0 13:55 pts/0 00:00:00 grep --color=auto firefox

root@server:~/linkdir# kill -9 3443

root@server:~/linkdir# ps -ef | grep firefox

root 3710 3388 0 13:57 pts/0 00:00:00 grep --color=auto firefox ⇐ firefox와 자식 프로세스가 모두 종료

root@server:~/linkdir# pstree

systemd─┬─MainThread─┬─Privileged Cont───20\*[{Privileged Cont}]

│ ├─Web Content───17\*[{Web Content}]

│ ├─WebExtensions───19\*[{WebExtensions}]

│ └─65\*[{MainThread}]

├─ModemManager───2\*[{ModemManager}]

├─NetworkManager───2\*[{NetworkManager}]

├─3\*[VBoxClient───VBoxClient───2\*[{VBoxClient}]]

├─VBoxClient───VBoxClient───3\*[{VBoxClient}]

├─VBoxService───8\*[{VBoxService}]

├─accounts-daemon───2\*[{accounts-daemon}]

├─acpid

├─avahi-daemon───avahi-daemon

├─boltd───2\*[{boltd}]

├─colord───2\*[{colord}]

├─cron

├─cups-browsed───2\*[{cups-browsed}]

├─cupsd

├─dbus-daemon

├─gdm3─┬─gdm-session-wor─┬─gdm-wayland-ses─┬─gnome-session-b─┬─gnome-sh+

│ │ │ │ ├─gsd-a11y+

│ │ │ │ ├─gsd-clip+

│ │ │ │ ├─gsd-colo+

│ │ │ │ ├─gsd-date+

│ │ │ │ ├─gsd-hous+

│ │ │ │ ├─gsd-keyb+

│ │ │ │ ├─gsd-medi+

│ │ │ │ ├─gsd-mous+

│ │ │ │ ├─gsd-powe+

│ │ │ │ ├─gsd-prin+

│ │ │ │ ├─gsd-rfki+

│ │ │ │ ├─gsd-scre+

│ │ │ │ ├─gsd-shar+

│ │ │ │ ├─gsd-smar+

│ │ │ │ ├─gsd-soun+

│ │ │ │ ├─gsd-waco+

│ │ │ │ ├─gsd-xset+

│ │ │ │ └─3\*[{gnom+

│ │ │ └─2\*[{gdm-wayland-ses}]

│ │ └─2\*[{gdm-session-wor}]

│ ├─gdm-session-wor─┬─gdm-x-session─┬─Xorg───{Xorg}

│ │ │ ├─gnome-session-b─┬─deja-dup-m+

│ │ │ │ ├─gnome-shel+

│ │ │ │ ├─gnome-soft+

│ │ │ │ ├─gsd-a11y-s+

│ │ │ │ ├─gsd-clipbo+

│ │ │ │ ├─gsd-color─+++

│ │ │ │ ├─gsd-dateti+

│ │ │ │ ├─gsd-disk-u+

│ │ │ │ ├─gsd-housek+

│ │ │ │ ├─gsd-keyboa+

│ │ │ │ ├─gsd-media-+

│ │ │ │ ├─gsd-mouse─+++

│ │ │ │ ├─gsd-power─+++

│ │ │ │ ├─gsd-print-+

│ │ │ │ ├─gsd-rfkill+++

│ │ │ │ ├─gsd-screen+

│ │ │ │ ├─gsd-sharin+

│ │ │ │ ├─gsd-smartc+

│ │ │ │ ├─gsd-sound─+++

│ │ │ │ ├─gsd-wacom─+++

│ │ │ │ ├─gsd-xsetti+

│ │ │ │ ├─nautilus-d+

│ │ │ │ ├─ssh-agent

│ │ │ │ ├─update-not+

│ │ │ │ └─3\*[{gnome-+

│ │ │ └─2\*[{gdm-x-session}]

│ │ └─2\*[{gdm-session-wor}]

│ └─2\*[{gdm3}]

├─gnome-keyring-d─┬─ssh-agent

│ └─3\*[{gnome-keyring-d}]

├─gsd-printer───2\*[{gsd-printer}]

├─2\*[ibus-x11───2\*[{ibus-x11}]]

├─2\*[kerneloops]

├─networkd-dispat───{networkd-dispat}

├─nginx───nginx

├─packagekitd───2\*[{packagekitd}]

├─polkitd───2\*[{polkitd}]

├─pulseaudio───3\*[{pulseaudio}]

├─rsyslogd───3\*[{rsyslogd}]

├─rtkit-daemon───2\*[{rtkit-daemon}]

├─snapd───14\*[{snapd}]

├─sshd───sshd───sshd

├─systemd───(sd-pam)

├─systemd─┬─(sd-pam)

│ ├─at-spi-bus-laun─┬─dbus-daemon

│ │ └─3\*[{at-spi-bus-laun}]

│ ├─at-spi2-registr───2\*[{at-spi2-registr}]

│ ├─dbus-daemon

│ ├─ibus-portal───2\*[{ibus-portal}]

│ ├─pulseaudio───3\*[{pulseaudio}]

│ └─xdg-permission-───2\*[{xdg-permission-}]

├─systemd─┬─(sd-pam)

│ ├─at-spi-bus-laun─┬─dbus-daemon

│ │ └─3\*[{at-spi-bus-laun}]

│ ├─at-spi2-registr───2\*[{at-spi2-registr}]

│ ├─dbus-daemon

│ ├─dconf-service───2\*[{dconf-service}]

│ ├─evolution-addre─┬─evolution-addre───5\*[{evolution-addre}]

│ │ └─4\*[{evolution-addre}]

│ ├─evolution-calen─┬─evolution-calen───8\*[{evolution-calen}]

│ │ └─4\*[{evolution-calen}]

│ ├─evolution-sourc───3\*[{evolution-sourc}]

│ ├─gnome-shell-cal───5\*[{gnome-shell-cal}]

│ ├─gnome-terminal-─┬─bash───sudo───su───bash───su───bash───sud+

│ │ └─3\*[{gnome-terminal-}]

│ ├─goa-daemon───3\*[{goa-daemon}]

│ ├─goa-identity-se───3\*[{goa-identity-se}]

│ ├─gvfs-afc-volume───3\*[{gvfs-afc-volume}]

│ ├─gvfs-goa-volume───2\*[{gvfs-goa-volume}]

│ ├─gvfs-gphoto2-vo───2\*[{gvfs-gphoto2-vo}]

│ ├─gvfs-mtp-volume───2\*[{gvfs-mtp-volume}]

│ ├─gvfs-udisks2-vo───2\*[{gvfs-udisks2-vo}]

│ ├─gvfsd─┬─gvfsd-trash───2\*[{gvfsd-trash}]

│ │ └─2\*[{gvfsd}]

│ ├─gvfsd-fuse───5\*[{gvfsd-fuse}]

│ ├─ibus-portal───2\*[{ibus-portal}]

│ └─xdg-permission-───2\*[{xdg-permission-}]

├─systemd-journal

├─systemd-logind

├─systemd-resolve

├─systemd-udevd

├─udisksd───4\*[{udisksd}]

├─unattended-upgr───{unattended-upgr}

├─upowerd───2\*[{upowerd}]

├─whoopsie───2\*[{whoopsie}]

└─wpa\_supplicant

잠시 쉬고 14시 15분에 이어서 진행하겠습니다.

# 실습 5-4 포그라운드 프로세스와 백그라운드 프로세스 (P195)

### #1 무한정 반복되는 프로세스를 실행

root@server:~# yes > /dev/null

### #2 ps 명령어를 이용해서 yes 프로세스의 실행 상태를 확인

ubuntu@server:~$ ps -ef | grep yes

root 3939 3388 94 14:17 pts/0 00:00:58 yes

ubuntu 3957 3943 0 14:18 pts/1 00:00:00 grep --color=auto yes

### #3 포그라운드 프로세스를 종료

ubuntu@server:~$ sudo kill -9 3939

또는

Ctl+C

### #4 포그라운드 프로세스를 백그라운드 프로세스로 변경(전이)

root@server:~# yes > /dev/null

^Z

[1]+ 정지됨 yes > /dev/null

root@server:~# bg

[1]+ yes > /dev/null & ⇐ 백그라운드로 실행되는 것을 표시

root@server:~# ps -ef | grep yes

root 3985 3388 69 14:23 pts/0 00:01:27 yes

root 3988 3388 0 14:25 pts/0 00:00:00 grep --color=auto yes

### #5 백그라운드 프로세스를 포그라운드 프로세스로 변경(전이)

root@server:~# jobs

[1]+ 실행중 yes > /dev/null & ⇐ Job ID (작업 번호) 확인

root@server:~# fg 1

yes > /dev/null ⇐ 포그라운드로 실행되는 것을 확인

### #6 gedit 프로그램을 포그라운드로 실행했을 때와 백그라운드로 실행했을 때 비교

root@server:~# gedit ⇐ 포그라운드로 실행한 것이고, gedit를 종료하기 전까지는 명령어를 추가로 실행할 수 없음

root@server:~# gedit & ⇐ 백그라운드에서 실행하는 것이고, gedit와 별개로 명령어 실행이 가능

[1] 4061

root@server:~# ls ⇐ gedit 실행 상태에서 명령어 실행이 가능

linkdir mydata.txt orginalfile

root@server:~# dpkg -l firefox

희망상태=알수없음(U)/설치(I)/지우기(R)/깨끗이(P)/고정(H)

| 상태=아님(N)/설치(I)/설정(C)/풀림(U)/절반설정(F)/일부설치(H)/트리거대기(W)/

| / 트리거밀림(T)

|/ 오류?=(없음)/다시설치필요(R) (상태, 오류가 대문자=불량)

||/ 이름 버전 Architecture 설명

+++-==============-============-============-=================================

ii firefox 80.0.1+build amd64 Safe and easy web browser from Mo

root@server:~#

# 실습 6-1 dpkg 명령어로 패키지 설치 (P209)

설치할 파일

http://archive.ubuntu.com/ubuntu/pool/universe/m/mc/mc\_4.8.19-1\_amd64.deb http://archive.ubuntu.com/ubuntu/pool/universe/n/ncftp/ncftp\_3.2.5-2\_amd64.deb

### #1 패키지 파일을 다운로드

root@server:~# wget http://archive.ubuntu.com/ubuntu/pool/universe/m/mc/mc\_4.8.19-1\_amd64.deb

--2020-09-08 14:52:01-- http://archive.ubuntu.com/ubuntu/pool/universe/m/mc/mc\_4.8.19-1\_amd64.deb

Resolving archive.ubuntu.com (archive.ubuntu.com)... 91.189.88.142, 91.189.88.152, 2001:67c:1360:8001::24, ...

접속 archive.ubuntu.com (archive.ubuntu.com)|91.189.88.142|:80... 접속됨.

HTTP request sent, awaiting response... 200 OK

Length: 473768 (463K) [application/x-debian-package]

Saving to: ‘mc\_4.8.19-1\_amd64.deb’

mc\_4.8.19-1\_amd64.d 100%[===================>] 462.66K 160KB/s in 2.9s

2020-09-08 14:52:05 (160 KB/s) - ‘mc\_4.8.19-1\_amd64.deb’ saved [473768/473768]

root@server:~# wget http://archive.ubuntu.com/ubuntu/pool/universe/n/ncftp/ncftp\_3.2.5-2\_amd64.deb

--2020-09-08 14:54:07-- http://archive.ubuntu.com/ubuntu/pool/universe/n/ncftp/ncftp\_3.2.5-2\_amd64.deb

Resolving archive.ubuntu.com (archive.ubuntu.com)... 91.189.88.142, 91.189.88.152, 2001:67c:1360:8001::23, ...

접속 archive.ubuntu.com (archive.ubuntu.com)|91.189.88.142|:80... 접속됨.

HTTP request sent, awaiting response... 200 OK

Length: 269736 (263K) [application/x-debian-package]

Saving to: ‘ncftp\_3.2.5-2\_amd64.deb’

ncftp\_3.2.5-2\_amd64 100%[===================>] 263.41K 177KB/s in 1.5s

2020-09-08 14:54:09 (177 KB/s) - ‘ncftp\_3.2.5-2\_amd64.deb’ saved [269736/269736]

root@server:~# ls \*.deb

mc\_4.8.19-1\_amd64.deb ncftp\_3.2.5-2\_amd64.deb

### #2 dpkg 명령어로 ncftp 설치

root@server:~# dpkg -l ncftp ⇐ 이미 설치되어 있는지 확인

dpkg-query: no packages found matching ncftp

root@server:~# dpkg --info ncftp\_3.2.5-2\_amd64.deb ⇐ 패키지 정보를 확인

new Debian package, version 2.0.

size 269736 bytes: control archive=1319 bytes.

588 bytes, 14 lines control

1194 bytes, 19 lines md5sums

201 bytes, 10 lines \* postinst #!/bin/sh

115 bytes, 13 lines \* prerm #!/bin/sh

Package: ncftp

Version: 2:3.2.5-2

Architecture: amd64

Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>

Original-Maintainer: Noël Köthe <noel@debian.org>

Installed-Size: 1526

Depends: libc6 (>= 2.15), libncurses5 (>= 6), libtinfo5 (>= 6)

Section: net

Priority: optional

Homepage: http://www.ncftpd.com/ncftp/

Description: User-friendly and well-featured FTP client

Ncftp allows a user to transfer files to and from a remote network

site, and offers additional features that are not found in the standard

interface, ftp. This version has Readline support enabled.

root@server:~# dpkg -i ncftp\_3.2.5-2\_amd64.deb ⇐ 설치

Selecting previously unselected package ncftp.

(데이터베이스 읽는중 ...현재 167498개의 파일과 디렉터리가 설치되어 있습니다.)

Preparing to unpack ncftp\_3.2.5-2\_amd64.deb ...

Unpacking ncftp (2:3.2.5-2) ...

ncftp (2:3.2.5-2) 설정하는 중입니다 ...

update-alternatives: using /usr/bin/ncftp3 to provide /usr/bin/ncftp (ncftp) in auto mode

Processing triggers for man-db (2.8.3-2ubuntu0.1) ...

root@server:~# dpkg -l ncftp ⇐ 설치 확인

희망상태=알수없음(U)/설치(I)/지우기(R)/깨끗이(P)/고정(H)

| 상태=아님(N)/설치(I)/설정(C)/풀림(U)/절반설정(F)/일부설치(H)/트리거대기(W)/

| / 트리거밀림(T)

|/ 오류?=(없음)/다시설치필요(R) (상태, 오류가 대문자=불량)

||/ 이름 버전 Architecture 설명

+++-==============-============-============-=================================

ii ncftp 2:3.2.5-2 amd64 User-friendly and well-featured F

root@server:~#

### #3 dpkg 명령어로 ncftp 삭제

root@server:~# dpkg -r ncftp

(데이터베이스 읽는중 ...현재 167517개의 파일과 디렉터리가 설치되어 있습니다.)

Removing ncftp (2:3.2.5-2) ...

Processing triggers for man-db (2.8.3-2ubuntu0.1) ...

### #4 의존성 문제가 있는 패키지를 설치

root@server:~# dpkg -i mc\_4.8.19-1\_amd64.deb

Selecting previously unselected package mc.

(데이터베이스 읽는중 ...현재 167498개의 파일과 디렉터리가 설치되어 있습니다.)

Preparing to unpack mc\_4.8.19-1\_amd64.deb ...

Unpacking mc (3:4.8.19-1) ...

dpkg: dependency problems prevent configuration of mc:

mc 패키지는 다음 패키지에 의존: libssh2-1 (>= 1.2.8): 하지만:

libssh2-1 패키지는 설치하지 않았습니다. ⇒ 의존성 문제가 발생하는 패키지를 먼저 설치

mc 패키지는 다음 패키지에 의존: mc-data (= 3:4.8.19-1): 하지만:

mc-data 패키지는 설치하지 않았습니다.

dpkg: error processing package mc (--install):

의존성 문제 - 설정하지 않고 남겨둠

Processing triggers for gnome-menus (3.13.3-11ubuntu1.1) ...

Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.2) ...

Processing triggers for mime-support (3.60ubuntu1) ...

처리하는데 오류가 발생했습니다:

mc

root@server:~# mc ⇐ 실행 시 의존성 오류 발생

mc: error while loading shared libraries: libssh2.so.1: cannot open shared object file: No such file or directory

잠시 쉬고, 15시 25분에 이어서 진행하겠습니다.

# 실습 6-2 apt-get 명령어로 mc 패키지 설치하기 (P215)

### #1 설치를 완료하지 못 한 mc 패키지를 삭제

root@server:~# dpkg -r mc

(데이터베이스 읽는중 ...현재 167597개의 파일과 디렉터리가 설치되어 있습니다.)

Removing mc (3:4.8.19-1) ...

Processing triggers for gnome-menus (3.13.3-11ubuntu1.1) ...

Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.2) ...

Processing triggers for mime-support (3.60ubuntu1) ...

### #2 설치할 패키지 정보를 확인

root@server:~# apt-cache show mc

Package: mc

Architecture: amd64

Version: 3:4.8.19-1

Priority: optional

Section: universe/utils

Origin: Ubuntu

Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>

Original-Maintainer: Debian MC Packaging Group <pkg-mc-devel@lists.alioth.debian.org>

Bugs: https://bugs.launchpad.net/ubuntu/+filebug

Installed-Size: 1452

Provides: mcedit

Depends: e2fslibs (>= 1.37), libc6 (>= 2.15), libglib2.0-0 (>= 2.35.9), libgpm2 (>= 1.20.4), libslang2 (>= 2.2.4), libssh2-1 (>= 1.2.8), mc-data (= 3:4.8.19-1)

Recommends: mime-support, perl, unzip

Suggests: arj, bzip2, catdvi | texlive-binaries, dbview, djvulibre-bin, file, genisoimage, gv, imagemagick, libaspell-dev, links | w3m | lynx, odt2txt, poppler-utils, python, python-boto, python-tz, xpdf | pdf-viewer, zip

Filename: pool/universe/m/mc/mc\_4.8.19-1\_amd64.deb

Size: 473768

MD5sum: 042652517a66c5e8f17149e26326564a

SHA1: 439660ea74c3c9fe0b6f448fb91b3b93a10cc0d4

SHA256: 494fe386a93862c523b6c6bab766f21e77717bc6601cb7af59960e67c5162a72

Homepage: https://www.midnight-commander.org

Description-en: Midnight Commander - a powerful file manager

GNU Midnight Commander is a text-mode full-screen file manager. It

uses a two panel interface and a subshell for command execution. It

includes an internal editor with syntax highlighting and an internal

viewer with support for binary files. Also included is Virtual

Filesystem (VFS), that allows files on remote systems (e.g. FTP, SSH

servers) and files inside archives to be manipulated like real files.

Description-md5: 252a5c5aeeb7425db45357d4ab8aa55f

### #3 의존성 정보를 확인

root@server:~# apt-cache depends mc

mc

의존: e2fslibs

libext2fs2

의존: libc6

의존: libglib2.0-0

의존: libgpm2

의존: libslang2

의존: libssh2-1

의존: mc-data

추천: mime-support

추천: perl

추천: unzip

unzip:i386

제안: arj

제안: bzip2

bzip2:i386

|제안: catdvi

제안: texlive-binaries

texlive-binaries:i386

제안: dbview

제안: djvulibre-bin

제안: file

file:i386

제안: genisoimage

제안: gv

제안: imagemagick

graphicsmagick-imagemagick-compat

imagemagick:i386

imagemagick-6.q16:i386

imagemagick-6.q16

제안: libaspell-dev

|제안: links

links:i386

|제안: w3m

w3m:i386

제안: lynx

lynx:i386

제안: odt2txt

제안: poppler-utils

poppler-utils:i386

제안: python

제안: python-boto

제안: python-tz

|제안: xpdf

제안: <pdf-viewer>

okular

atril

evince

gv

mupdf

viewpdf.app

xpdf

zathura-pdf-poppler

제안: zip

zip:i386

### #4 패키지 설치

root@server:~# apt-get install mc

패키지 목록을 읽는 중입니다... 완료

의존성 트리를 만드는 중입니다

상태 정보를 읽는 중입니다... 완료

다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다:

efibootmgr libfwup1 libllvm9

Use 'sudo apt autoremove' to remove them.

다음의 추가 패키지가 설치될 것입니다 :

libssh2-1 mc-data

제안하는 패키지:

arj catdvi | texlive-binaries dbview djvulibre-bin gv libaspell-dev links

| w3m | lynx odt2txt python python-boto python-tz

다음 새 패키지를 설치할 것입니다:

libssh2-1 mc mc-data

0개 업그레이드, 3개 새로 설치, 0개 제거 및 3개 업그레이드 안 함.

1,785 k바이트 아카이브를 받아야 합니다.

이 작업 후 7,541 k바이트의 디스크 공간을 더 사용하게 됩니다.

계속 하시겠습니까? [Y/n]

받기:1 http://kr.archive.ubuntu.com/ubuntu bionic/universe amd64 libssh2-1 amd64 1.8.0-1 [73.2 kB]

받기:2 http://kr.archive.ubuntu.com/ubuntu bionic/universe amd64 mc-data all 3:4.8.19-1 [1,238 kB]

받기:3 http://kr.archive.ubuntu.com/ubuntu bionic/universe amd64 mc amd64 3:4.8.19-1 [474 kB]

내려받기 1,785 k바이트, 소요시간 7초 (250 k바이트/초)

Selecting previously unselected package libssh2-1:amd64.

(데이터베이스 읽는중 ...현재 167507개의 파일과 디렉터리가 설치되어 있습니다.)

Preparing to unpack .../libssh2-1\_1.8.0-1\_amd64.deb ...

Unpacking libssh2-1:amd64 (1.8.0-1) ...

Selecting previously unselected package mc-data.

Preparing to unpack .../mc-data\_3%3a4.8.19-1\_all.deb ...

Unpacking mc-data (3:4.8.19-1) ...

Selecting previously unselected package mc.

Preparing to unpack .../mc\_3%3a4.8.19-1\_amd64.deb ...

Unpacking mc (3:4.8.19-1) ...

mc-data (3:4.8.19-1) 설정하는 중입니다 ...

libssh2-1:amd64 (1.8.0-1) 설정하는 중입니다 ...

mc (3:4.8.19-1) 설정하는 중입니다 ...

update-alternatives: using /usr/bin/mcview to provide /usr/bin/view (view) in auto mode

Processing triggers for gnome-menus (3.13.3-11ubuntu1.1) ...

Processing triggers for hicolor-icon-theme (0.17-2) ...

Processing triggers for mime-support (3.60ubuntu1) ...

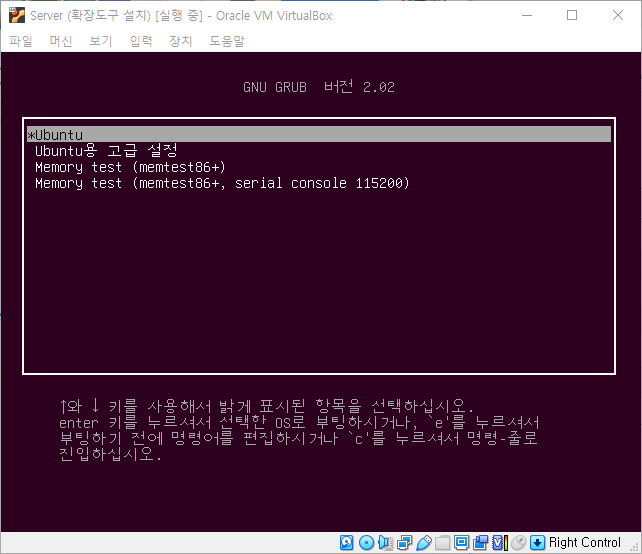
Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.2) ...

Processing triggers for libc-bin (2.27-3ubuntu1.2) ...

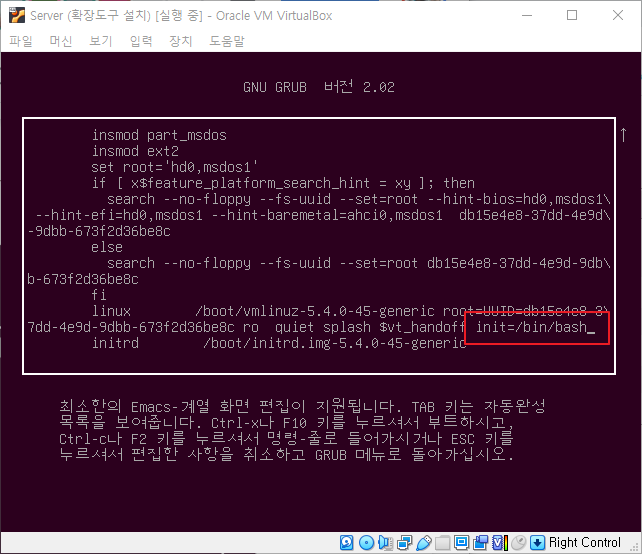
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...

# 실습 6-4 root 사용자의 비밀번호 분실 시 조치하기

부팅 시 ESC 키를 누릅니다.



E 키를 눌러서 편집 모드로 진입 > init=/bin/bash 를 추가 > F10 키를 눌러서 재시작



잠시 쉬고 16시 25분에 이어서 진행하겠습니다.

## 쉘에서 변수 사용법

ubuntu@server-b:~$ a=10

ubuntu@server-b:~$ echo $a

10

ubuntu@server-b:~$ b=20

ubuntu@server-b:~$ echo $b

20

ubuntu@server-b:~$ echo $a+$b

10+20

## 앞에서 했던 작업을 파일로 만들어서 일괄로 실행

ubuntu@server-b:~$ vi vartest.sh

|  |
| --- |
| #!/bin/bash  a=10  echo $a  b=20  echo $b  echo $a+$b |

ubuntu@server-b:~$ chmod +x vartest.sh

ubuntu@server-b:~$ ./vartest.sh

10

20

10+20

ubuntu@server-b:~$

## 변수를 나타내는 $ 기호를 이스케이프시키는 방법과 사용자 입력을 변수로 할당하는 방법

ubuntu@server-b:~$ vi var1.sh

|  |
| --- |
| #!/bin/bash  myvar="Hi Woo" ⇐ (0)  echo $myvar ⇐ (1)  echo "$myvar" ⇐ (2)  echo '$myvar' ⇐ (3)  echo \$myvar ⇐ (4)  echo 값 입력:  read myvar ⇐ (5) 사용자가 입력한 값이 저장  echo '$myvar' = $myvar  exit 0 |

ubuntu@server-b:~$ sh var1.sh

Hi Woo ⇐ (1) $myvar를 변수로 인식해서 (0)에서 정의한 변수의 값이 출력

Hi Woo ⇐ (2)

$myvar ⇐ (3) $myvar를 변수로 인식하지 않고 문자열로 인식 ⇒ 이스케이프되었다

$myvar ⇐ (4)

값 입력:

abcd ⇐ 입력한 값

$myvar = abcd

## 숫자 연산

ubuntu@server-b:~$ vi numcalc.sh

|  |
| --- |
| #!/bin/sh  num1=100  num2=$num1+200  echo $num2  num3=`expr $num1 + 200`  echo $num3  num4=`expr \( $num1 + 200 \) / 10 \\* 2`  echo $num4  exit 0 |

ubuntu@server-b:~$ sh numcalc.sh

100+200

300

60

## 파라미터 변수

ubuntu@server-b:~$ vi paravar.sh

|  |
| --- |
| #!/bin/sh  echo "실행파일 이름 : <$0>"  echo "첫번째 파라미터 : <$1>, 두번째 파라미터 : <$2>"  echo "전체 파라미터 : <$\*>"  exit 0 |

ubuntu@server-b:~$ sh paravar.sh 11111 22222 33333 44444

실행파일 이름 : <paravar.sh>

첫번째 파라미터 : <11111>, 두번째 파라미터 : <22222>

전체 파라미터 : <11111 22222 33333 44444>

쉘 스크립트 작성 및 실행, … 반드시 복습, 예습, 과외, … 해 보세요.

9/9 (수)

# 쉘 스크립트에서 for 루프 사용 방법

root@server:~# vi sigma.sh

|  |
| --- |
| #!/bin/sh  hap=0  for i in 1 2 3 4 5 6 7 8 9 10  do  echo $i  hap=`expr $hap + $i`  done  echo "1부터 10까지의 합은 " $hap  exit 0 |

1부터 100까지의 합을 구하는 쉘 스크립트를 작성하시오.

root@server:~# vi sigma.sh

|  |
| --- |
| #!/bin/sh  hap=0  #for i in 1 2 3 4 5 6 7 8 9 10  for i in $(seq 1 100)  do  hap=`expr $hap + $i`  done  echo "1부터 100까지의 합은 " $hap  exit 0 |

root@server:~# vi sigma.sh

|  |
| --- |
| #!/bin/sh  hap=0  #for i in 1 2 3 4 5 6 7 8 9 10  #for i in $(seq 1 100)  for i in {1..100}  do  hap=`expr $hap + $i`  done  echo "1부터 100까지의 합은 " $hap  exit 0 |

root@server:~# sh sigma.sh

expr: non-integer argument

1부터 100까지의 합은

root@server:~# bash sigma.sh

1부터 100까지의 합은 5050

root@server:~# vi sigma.sh

|  |
| --- |
| #!/bin/sh  hap=0  #for i in 1 2 3 4 5 6 7 8 9 10  #for i in $(seq 1 100)  #for i in {1..100}  for (( i = 1 ; i <= 100 ; i ++ ))  do  hap=`expr $hap + $i`  done  echo "1부터 100까지의 합은 " $hap  exit 0 |

root@server:~# sh sigma.sh

sigma.sh: 7: sigma.sh: Syntax error: Bad for loop variable

root@server:~# bash sigma.sh

1부터 100까지의 합은 5050

잠깐 쉬고, 10시 10분에 이어서 진행하겠습니다.

Quiz.

1부터 10까지 숫자 중 짝수의 합을 구하시오.

|  |
| --- |
| #!/bin/sh  hap=0  for i in 2 4 6 8 10 ⇐ 짝수만 나열  #for i in $(seq 2 2 10) ⇐ seq 시작 증가치 끝  #for i in {2..10..2} ⇐ {시작..끝..증가치}  #for (( i = 2 ; i <= 10 ; i += 2 ))  do  hap=`expr $hap + $i`  done  echo "1부터 10까지의 짝수 합은 " $hap  exit 0 |

Quiz.

아래와 같은 형식으로 구구단을 출력하시오.

2 x 1 = 2

2 x 2 = 4

:

2 x 9 = 18

-----------

3 x 2 = 3

:

------------

:

9 x 9 = 81

-----------

root@server:~# vi gugudan.sh

|  |
| --- |
| #!/bin/bash  for i in $(seq 1 9)  do  for j in {1..9}  do  # echo $i " x " $j " = " `expr $i \\* $j`  echo $i x $j = `expr $i \\* $j`  done  echo "--------------"  done  exit 0 |

root@server:~# chmod +x gugudan.sh

root@server:~# ./gugudan.sh

Quiz.

아래와 같은 형식으로 구구단을 출력하시오.

2 x 1 = 2 3 x 1 = 3 .. 9 x 1 = 9

: : :

2 x 9 = 18 3 x 9 = 27 .. 9 x 9 = 81

Hint.

printf "%s x %s = %s \t" $i $j $k

~~~~~~~~~~~~

형식에 맞춰서 변수 i, j, k 값을 출력

만약, i, j, k 값이 9, 8, 72 이면, 9 x 8 = 72 형태로 출력

|  |
| --- |
| #!/bin/bash  for j in $(seq 1 9)  do  for i in {1..9}  do  printf "%s x %s = %s\t" $i $j `expr $i \\* $j`  if [ $i = 9 ]  then  printf "\n"  fi  done  done  exit 0 |

root@server:~# ./gugudan.sh

1 x 1 = 1 2 x 1 = 2 3 x 1 = 3 4 x 1 = 4 ... 8 x 1 = 8 9 x 1 = 9

1 x 2 = 2 2 x 2 = 4 3 x 2 = 6 4 x 2 = 8 ... 8 x 2 = 16 9 x 2 = 18

1 x 3 = 3 2 x 3 = 6 3 x 3 = 9 4 x 3 = 12 ... 8 x 3 = 24 9 x 3 = 27

1 x 4 = 4 2 x 4 = 8 3 x 4 = 12 4 x 4 = 16 ... 8 x 4 = 32 9 x 4 = 36

1 x 5 = 5 2 x 5 = 10 3 x 5 = 15 4 x 5 = 20 ... 8 x 5 = 40 9 x 5 = 45

1 x 6 = 6 2 x 6 = 12 3 x 6 = 18 4 x 6 = 24 ... 8 x 6 = 48 9 x 6 = 54

1 x 7 = 7 2 x 7 = 14 3 x 7 = 21 4 x 7 = 28 ... 8 x 7 = 56 9 x 7 = 63

1 x 8 = 8 2 x 8 = 16 3 x 8 = 24 4 x 8 = 32 ... 8 x 8 = 64 9 x 8 = 72

1 x 9 = 9 2 x 9 = 18 3 x 9 = 27 4 x 9 = 36 ... 8 x 9 = 72 9 x 9 = 81

**작성한 gugudan.sh 파일 내용을 아래 주소에 OO조\_OOO 탭을 만들고 코드를 붙여넣어 주세요.**

https://docs.google.com/spreadsheets/d/1iKvOyE5Gq7q1E4wgrqcSiPe6KWA7G2FLTX2MVGAfOPQ/edit?usp=sharing

Quiz.

아래 조건을 만족하는 quiz.sh 프로그램을 작성하시오.

1. 프로그램에서 임의의 숫자를 생성 (rand 명령어를 이용) ⇐ **apt install rand** 명령어로 rand 설치
2. 사용자로부터 숫자를 입력받아서 1)에서 생성한 숫자를 맞추는 프로그램
3. 만약, 사용자가 입력한 숫자가 1)에서 생성한 숫자와 다르면, 크다, 작다 메시지를 출력
4. 사용자가 입력한 숫자가 1)에서 생성한 숫자와 일치하면 맞다 메시지를 출력
5. 단, 사용자가 숫자를 입력하는 회수는 10회로 제한하고, 10회를 초과하는 경우 실패 메시지를 출력

**작성한 quiz.sh 파일 내용을 아래 주소에 OO조\_OOO 탭을 만들고 코드를 붙여넣어 주세요.**

https://docs.google.com/spreadsheets/d/1D58nS6csvXf-RvrmTnnsRk4vBdmioSUrLbMlD1SDh\_A/edit?usp=sharing

#!/bin/bash

rnum=$(rand) ⇐ rand 명령어의 실행 결과를 rnum 변수에 할당

if [ $# -gt 0 ] && [ $1 = "debug" ] ⇐ 쉘 스크립트로 전달된 파라미터의 개수가 1개 이상이고,

then 첫번째 파라미터의 값이 debug인 경우, 생성한 랜덤 넘버를 출력

echo $rnum

fi

for (( i = 0 ; i < 10 ; i ++ )) ⇐ 10번 반복 (사용자 입력과 값 비교)

do

echo "Input number :"

read inum ⇐ 사용자 입력을 처리

if [ $inum -eq $rnum ] ⇐ 값이 일치하는지 비교

then

echo "맞다"

exit 0

elif [ $inum -lt $rnum ] ⇐ 값이 작은 경우

then

echo "작다"

else ⇐ 값이 큰 경우

echo "크다"

fi

done

echo "실패" ⇐ 10번 반복했는데도 일치하지 않는 경우

exit 0

root@server:~# chmod +x quiz.sh

root@server:~# ./quiz.sh

root@server:~# ./quiz.sh debug ⇐ 디버깅 목적으로 프로그램을 실행 → 랜덤 넘버를 화면에 제공

점심 식사하고, 13시 10분에 이어서 진행하겠습니다.

구구단 가로 출력과 퀴즈 2개를 각각 구글 문서에 등록해 주세요.

등록 주소가 다릅니다. 확인하고 등록해 주세요.

가급적이면 보기 좋게 하나의 쉘에 넣어 주세요.

root@server:~# vi shift.sh

#!/bin/bash

myfunc() {

str=""

while [ "$1" != "" ]; do

str="$str $1"

shift

done

echo $str

}

myfunc AAA BBB CCC DDD EEE FFF GGG

exit 0

## 크론(cron) 등록

crontab -l ⇐ 등록된 크론을 확인

crontab -e ⇐ 크론을 등록, 수정

0 5 \* \* 1 tar -zcf /var/backups/home.tgz /home/

분 시 일 월 요일

0~6 : 일~토

## 특정 시간에 특정 디렉터리를 백업

root@server:~# vi backup.sh

root@server:~# cat backup.sh

#!/bin/bash

set $(date)

fname="backup$1$2$3tar.xz" ⇐ 2020.09.09.tar.xz

tar cfJ /backup/$fname /home

root@server:~# mkdir /backup

root@server:~# bash ./backup.sh

tar: Removing leading `/' from member names

root@server:~# ls /backup/

backup2020.09.09.tar.xz

크론에 등록

root@server:~# vi /etc/crontab

# /etc/crontab: system-wide crontab

# Unlike any other crontab you don't have to run the `crontab'

# command to install the new version when you edit this file

# and files in /etc/cron.d. These files also have username fields,

# that none of the other crontabs do.

SHELL=/bin/sh

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user command

17 \* \* \* \* root cd / && run-parts --report /etc/cron.hourly

25 6 \* \* \* root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )

47 6 \* \* 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )

52 6 1 \* \* root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )

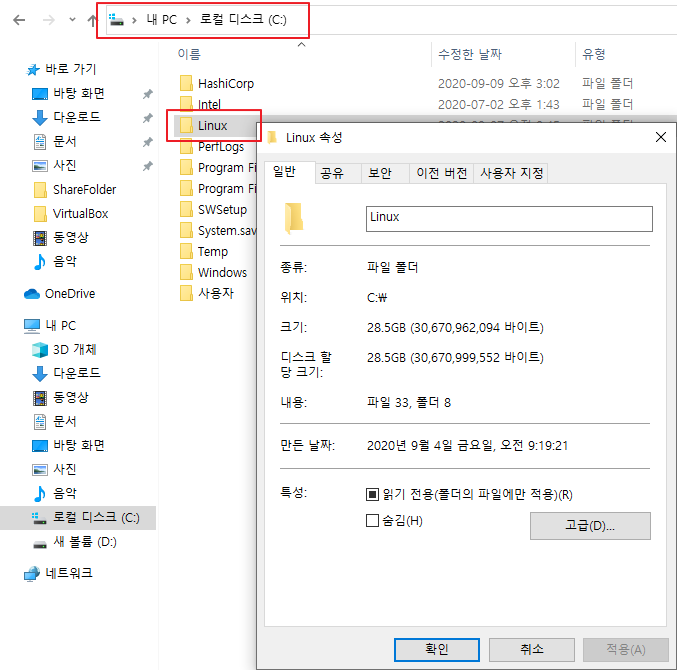
# 매월 16일 새벽 3시 20분에 백업을 수행

20 03 16 \* \* root /root/backup.sh

#

잠시 쉬고 14시 25분에 이어서 진행하겠습니다. (데브옵스 교재 준비해 주세요.)

사용하지 않는 가상머신 이미지가 있으면 삭제



프로젝트를 수행할 때 절차, 방법, 도구, 산출물 등을 정의하고 있는 것

⇒ 방법론(method)

https://sasperger.tistory.com/136

⇒ 애자일SW개발101 (PDF 파일)

# Vagrant로 로컬 개발 환경의 Infrastructure as Code 화 (P58)

Vagrant ⇒ 해시코드에서 제공하는 **가상 환경 구축 도구**

VirtualBox 6.1 버전을 사용하기 위해서는 Vagrant 2.2.1 보다 높은 버전을 사용해야 함

### #1 기존에 실행 중 가상머신을 모두 중지 (poweroff)

### #2 Vagrant 설치

https://www.vagrantup.com/ 환경에 맞는 설치 파일을 다운로드 받아서 설치

설치하고 리부팅 후 다시 접속해 주세요. 잠시 쉬고 15시 20분에 다시 시작하겠습니다.

### #3 작업 디렉터리를 생성

C:\> mkdir C:\HashiCorp\WorkDir

### #4 Vagrantfile 파일을 생성

C:\HashiCorp\WorkDir> vagrant init ⇐ Vagrantfile 템플릿을 생성

A `Vagrantfile` has been placed in this directory. You are now

ready to `vagrant up` your first virtual environment! Please read

the comments in the Vagrantfile as well as documentation on

`vagrantup.com` for more information on using Vagrant.

C:\HashiCorp\WorkDir> dir

C 드라이브의 볼륨에는 이름이 없습니다.

볼륨 일련 번호: 7681-638B

C:\HashiCorp\WorkDir 디렉터리

2020-09-09 오후 03:43 <DIR> .

2020-09-09 오후 03:43 <DIR> ..

2020-09-09 오후 03:43 <DIR> .vagrant

2020-09-09 오후 03:43 3,080 Vagrantfile ⇐ 생성된 것을 확인

1개 파일 3,080 바이트

3개 디렉터리 113,129,132,032 바이트 남음

### #5 Vagrantfile을 편집

|  |
| --- |
| # -\*- mode: ruby -\*-  # vi: set ft=ruby :  Vagrant.configure("2") do |config|  # config.vm.box = "centos/7"  config.vm.box = "generic/centos7"  config.vm.hostname = "demo"  config.vm.network "private\_network", ip: "192.168.33.10"  config.vm.synced\_folder ".", "/home/vagrant/sync", disabled: true  end |

### #6 가상머신을 생성하고 기동

C:\HashiCorp\WorkDir> vagrant up

Bringing machine 'default' up with 'virtualbox' provider...

==> default: Box 'generic/centos7' could not be found. Attempting to find and install...

default: Box Provider: virtualbox

default: Box Version: >= 0

==> default: Loading metadata for box 'generic/centos7'

default: URL: https://vagrantcloud.com/generic/centos7

==> default: Adding box 'generic/centos7' (v3.0.30) for provider: virtualbox

default: Downloading: https://vagrantcloud.com/generic/boxes/centos7/versions/3.0.30/providers/virtualbox.box

==> default: Box download is resuming from prior download progress

Download redirected to host: vagrantcloud-files-production.s3.amazonaws.com

default:

default: Calculating and comparing box checksum...

==> default: Successfully added box 'generic/centos7' (v3.0.30) for 'virtualbox'!

==> default: Importing base box 'generic/centos7'...

==> default: Matching MAC address for NAT networking...

==> default: Checking if box 'generic/centos7' version '3.0.30' is up to date...

==> default: Setting the name of the VM: WorkDir\_default\_1599636778803\_68703

==> default: Clearing any previously set network interfaces...

==> default: Preparing network interfaces based on configuration...

default: Adapter 1: nat

default: Adapter 2: hostonly

==> default: Forwarding ports...

default: 22 (guest) => 2222 (host) (adapter 1)

==> default: Running 'pre-boot' VM customizations...

==> default: Booting VM...

==> default: Waiting for machine to boot. This may take a few minutes...

default: SSH address: 127.0.0.1:2222

default: SSH username: vagrant

default: SSH auth method: private key

default: Warning: Connection aborted. Retrying...

default: Warning: Connection reset. Retrying...

default: Warning: Connection aborted. Retrying...

default: Warning: Connection reset. Retrying...

default: Warning: Connection aborted. Retrying...

default:

default: Vagrant insecure key detected. Vagrant will automatically replace

default: this with a newly generated keypair for better security.

default:

default: Inserting generated public key within guest...

default: Removing insecure key from the guest if it's present...

default: Key inserted! Disconnecting and reconnecting using new SSH key...

==> default: Machine booted and ready!

==> default: Checking for guest additions in VM...

default: The guest additions on this VM do not match the installed version of

default: VirtualBox! In most cases this is fine, but in rare cases it can

default: prevent things such as shared folders from working properly. If you see

default: shared folder errors, please make sure the guest additions within the

default: virtual machine match the version of VirtualBox you have installed on

default: your host and reload your VM.

default:

default: Guest Additions Version: 5.2.44

default: VirtualBox Version: 6.1

==> default: Setting hostname...

==> default: Configuring and enabling network interfaces...

C:\HashiCorp\WorkDir>

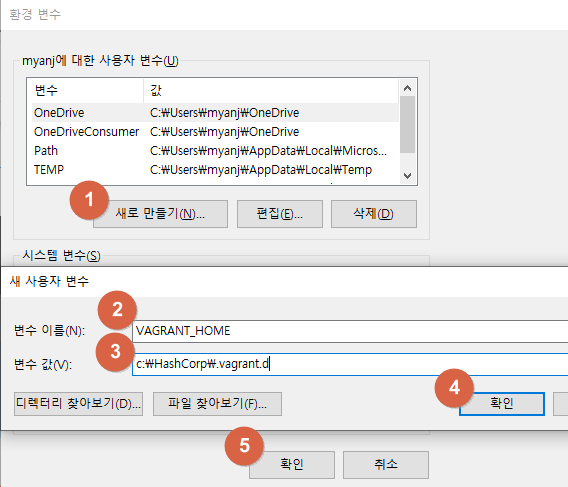
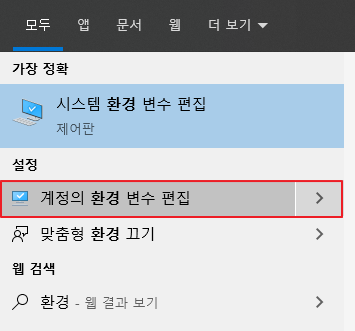


### Encoding::CompatibilityError: incompatible character encodings: UTF-8 and CP-949 오류가 발생하는 경우

원인 → 윈도우 사용자 계정명에 한글이 포함된 경우

조치

1. 사용자 홈 디렉터리(C:\Users\사용자명) 아래에 있는 .vagrant.d 폴더를 한글이 없는 디렉터리(예: C:\HashCorp)로 복사
2. 경로 환경 변수에 VAGRANT\_HOME 변수를 추가하고 1)에서 설정한 경로를 지정



1. **새로운 명령어 창을 실행**해야 변경된 정보가 반영

잠시 쉬고 16시 45분에 이어서 진행하겠습니다.

### #7 vagrant up 을 통해서 생성한 가상 머신으로 접속

### #7-1 방법1. VirtualBox를 이용



계정 : vagrant / vagrant

### #7-2 방법2. vagrant ssh 이용 ⇒ 별도의 프로그램 설치나 인증이 필요

C:\HashiCorp\WorkDir> vagrant ssh

Last login: Wed Sep 9 07:52:28 2020

[vagrant@demo ~]$ ip a

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether 08:00:27:52:0f:fe brd ff:ff:ff:ff:ff:ff

inet 10.0.2.15/24 brd 10.0.2.255 scope global noprefixroute dynamic eth0

valid\_lft 85216sec preferred\_lft 85216sec

3: eth1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether 08:00:27:32:e4:d4 brd ff:ff:ff:ff:ff:ff

inet 192.168.33.10/24 brd 192.168.33.255 scope global noprefixroute eth1

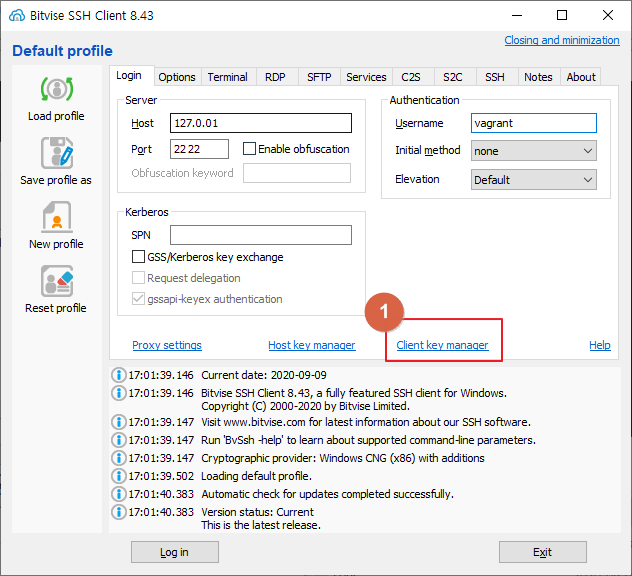
valid\_lft forever preferred\_lft forever

inet6 fe80::a00:27ff:fe32:e4d4/64 scope link

valid\_lft forever preferred\_lft forever

[vagrant@demo ~]$

### #7-3 방법3. SSH 클라이언트를 이용한 접속



키 파일이 있는 위치는 아래 명령어로 확인이 가능

C:\HashiCorp\WorkDir> vagrant ssh-config

Host default

HostName 127.0.0.1

User vagrant

Port 2222

UserKnownHostsFile /dev/null

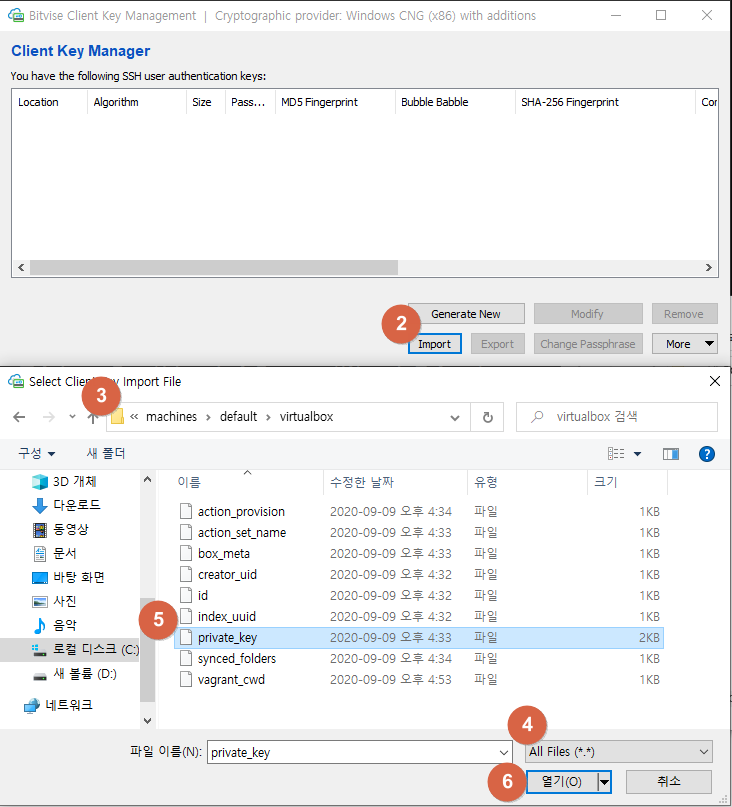
StrictHostKeyChecking no

PasswordAuthentication no

IdentityFile C:/HashiCorp/WorkDir/.vagrant/machines/default/virtualbox/private\_key

IdentitiesOnly yes

LogLevel FATAL



[vagrant@demo ~]$ ip a

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether 08:00:27:52:0f:fe brd ff:ff:ff:ff:ff:ff

inet 10.0.2.15/24 brd 10.0.2.255 scope global noprefixroute dynamic eth0

valid\_lft 84286sec preferred\_lft 84286sec

3: eth1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000

link/ether 08:00:27:32:e4:d4 brd ff:ff:ff:ff:ff:ff

inet 192.168.33.10/24 brd 192.168.33.255 scope global noprefixroute eth1

valid\_lft forever preferred\_lft forever

inet6 fe80::a00:27ff:fe32:e4d4/64 scope link

valid\_lft forever preferred\_lft forever

[vagrant@demo ~]$ whoami

vagrant

[vagrant@demo ~]$ uname -n

demo

### #8 도움말을 참조해서 스냅샷 생성

C:\HashiCorp\WorkDir> vagrant --help

Usage: vagrant [options] <command> [<args>]

-h, --help Print this help.

Common commands:

autocomplete manages autocomplete installation on host

box manages boxes: installation, removal, etc.

cloud manages everything related to Vagrant Cloud

destroy stops and deletes all traces of the vagrant machine

global-status outputs status Vagrant environments for this user

halt stops the vagrant machine

help shows the help for a subcommand

init initializes a new Vagrant environment by creating a Vagrantfile

login

package packages a running vagrant environment into a box

plugin manages plugins: install, uninstall, update, etc.

port displays information about guest port mappings

powershell connects to machine via powershell remoting

provision provisions the vagrant machine

push deploys code in this environment to a configured destination

rdp connects to machine via RDP

reload restarts vagrant machine, loads new Vagrantfile configuration

resume resume a suspended vagrant machine

snapshot manages snapshots: saving, restoring, etc.

ssh connects to machine via SSH

ssh-config outputs OpenSSH valid configuration to connect to the machine

status outputs status of the vagrant machine

suspend suspends the machine

up starts and provisions the vagrant environment

upload upload to machine via communicator

validate validates the Vagrantfile

version prints current and latest Vagrant version

winrm executes commands on a machine via WinRM

winrm-config outputs WinRM configuration to connect to the machine

For help on any individual command run `vagrant COMMAND -h`

Additional subcommands are available, but are either more advanced

or not commonly used. To see all subcommands, run the command

`vagrant list-commands`.

--[no-]color Enable or disable color output

--machine-readable Enable machine readable output

-v, --version Display Vagrant version

--debug Enable debug output

--timestamp Enable timestamps on log output

--debug-timestamp Enable debug output with timestamps

--no-tty Enable non-interactive output

C:\HashiCorp\WorkDir> vagrant snapshot --help

Usage: vagrant snapshot <subcommand> [<args>]

Available subcommands:

delete

list

pop

push

restore

save

For help on any individual subcommand run `vagrant snapshot <subcommand> -h`

--[no-]color Enable or disable color output

--machine-readable Enable machine readable output

-v, --version Display Vagrant version

--debug Enable debug output

--timestamp Enable timestamps on log output

--debug-timestamp Enable debug output with timestamps

--no-tty Enable non-interactive output

C:\HashiCorp\WorkDir> vagrant snapshot save -h

Usage: vagrant snapshot save [options] [vm-name] <name>

Take a snapshot of the current state of the machine. The snapshot

can be restored via `vagrant snapshot restore` at any point in the

future to get back to this exact machine state.

If no vm-name is given, Vagrant will take a snapshot of

the entire environment with the same snapshot name.

Snapshots are useful for experimenting in a machine and being able

to rollback quickly.

-f, --force Replace snapshot without confirmation

--[no-]color Enable or disable color output

--machine-readable Enable machine readable output

-v, --version Display Vagrant version

--debug Enable debug output

--timestamp Enable timestamps on log output

--debug-timestamp Enable debug output with timestamps

--no-tty Enable non-interactive output

-h, --help Print this help

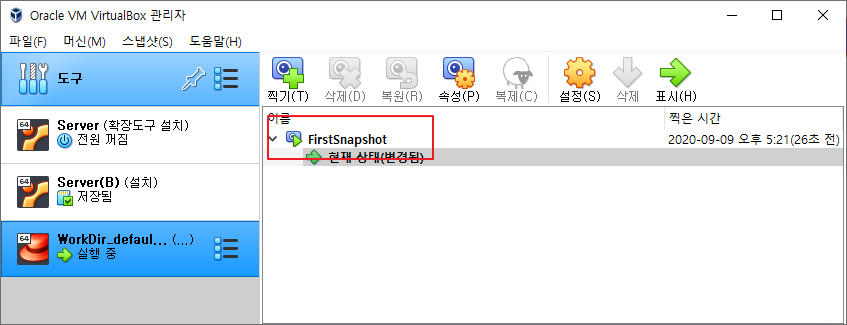
C:\HashiCorp\WorkDir> vagrant snapshot save FirstSnapshot

==> default: Snapshotting the machine as 'FirstSnapshot'...

==> default: Snapshot saved! You can restore the snapshot at any time by

==> default: using `vagrant snapshot restore`. You can delete it using

==> default: `vagrant snapshot delete`.



### #9 가상 머신을 정지 및 삭제

C:\HashiCorp\WorkDir> vagrant halt

==> default: Attempting graceful shutdown of VM...

C:\HashiCorp\WorkDir> vagrant destroy

default: Are you sure you want to destroy the 'default' VM? [y/N] y

==> default: Destroying VM and associated drives...

# Vagrant를 이용해서 팀 전체가 웹 서버가 설치된 동일한 가상 머신 환경을 공유 (P66)

https://www.vagrantup.com/docs/provisioning

### #1 Vagrantfile에 provisioning 내용을 추가

C:\HashiCorp\WorkDir\Vagrantfile

|  |
| --- |
| # -\*- mode: ruby -\*-  # vi: set ft=ruby :  Vagrant.configure("2") do |config|  # config.vm.box = "centos/7"  config.vm.box = "generic/centos7"  config.vm.hostname = "demo"  config.vm.network "private\_network", ip: "192.168.33.10"  config.vm.synced\_folder ".", "/home/vagrant/sync", disabled: true  config.vm.provision "shell", inline: $script  end  $script = <<SCRIPT  yum install -y epel-release  yum install -y nginx  echo "Hello, Vagrant" > /usr/share/nginx/html/index.html  systemctl start nginx  SCRIPT |

### #2 provisioning 실행

C:\HashiCorp\WorkDir> vagrant up

Bringing machine 'default' up with 'virtualbox' provider...

==> default: Importing base box 'generic/centos7'...

==> default: Matching MAC address for NAT networking...

==> default: Checking if box 'generic/centos7' version '3.0.30' is up to date...

==> default: Setting the name of the VM: WorkDir\_default\_1599640437410\_82346

==> default: Clearing any previously set network interfaces...

==> default: Preparing network interfaces based on configuration...

default: Adapter 1: nat

default: Adapter 2: hostonly

==> default: Forwarding ports...

default: 22 (guest) => 2222 (host) (adapter 1)

==> default: Running 'pre-boot' VM customizations...

==> default: Booting VM...

:

==> default: Setting hostname...

==> default: Configuring and enabling network interfaces...

==> default: Running provisioner: shell...

default: Running: inline script

default: Loaded plugins: fastestmirror

default: Determining fastest mirrors

default: \* base: mirror.kakao.com

default: \* epel: ftp.jaist.ac.jp

default: \* extras: mirror.kakao.com

default: \* updates: mirror.kakao.com

default: Package epel-release-7-12.noarch already installed and latest version

default: Nothing to do

default: Loaded plugins: fastestmirror

default: Loading mirror speeds from cached hostfile

default: \* base: mirror.kakao.com

default: \* epel: ftp.jaist.ac.jp

default: \* extras: mirror.kakao.com

default: \* updates: mirror.kakao.com

:

C:\HashiCorp\WorkDir>

### #3 결과를 확인

C:\HashiCorp\WorkDir> vagrant ssh ⇐ 호스트 PC에서 가신머신으로 SSH로 접속

[vagrant@demo ~]$ cat /usr/share/nginx/html/index.html ⇐ nginx 설치 여부 및 index.html 파일 수정 여부를 확인

Hello, Vagrant

[vagrant@demo ~]$ sudo systemctl stop firewalld ⇐ 방화벽 해제

[vagrant@demo ~]$ exit ⇐ 가상머신에 빠져 나옴

logout

Connection to 127.0.0.1 closed.

C:\HashiCorp\WorkDir> curl http://192.168.33.10 ⇐ 가상머신(nginx서버)에게 웹 페이지(index.html)를 요청

Hello, Vagrant ⇐ index.html 파일의 내용을 반환

### curl은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는 배치 파일이 아닙니다 오류가 발생하는 경우

⇒ 호스트 PC의 브라우저에서 주소창에 http://192.168.33.10 으로 접속했을 때 Hello, Vagrant가 출력되는지 확인

### 가상 머신이 기동된 후에 provison에 기술된 내용을 실행할 경우

> vagrant provision

또는

> vagrant reload --provision

교재 59페이지 ~ 68페이지 내용과 함께 복습해 보세요.

9/10 (목)

# Windows에서 curl 명령어 안 되는 경우

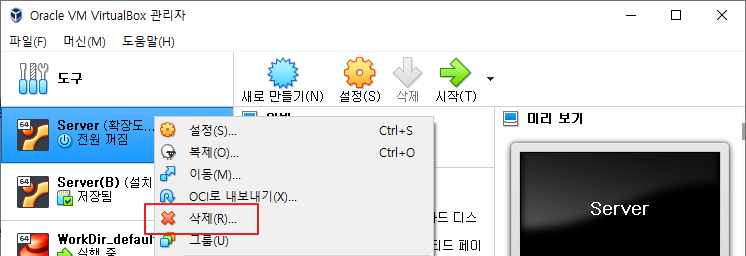
### #1 아래 URL에서 curl 프로그램을 다운로드 받아서 압축해제

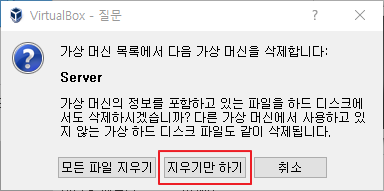
https://curl.haxx.se/windows/

### #2 압축해제한 폴더에 curl.exe 파일을 c:\windows\system32 폴더 아래로 복사

# C:\Linux 아래에 만들어 놓은 가상머신 이미지 파일들을 다른 폴더 또는 PC로 옮길 경우

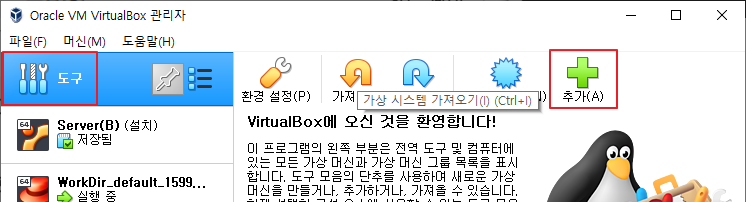
1. 가상머신을 종료
2. VirtualBox에서 가상머신을 제거



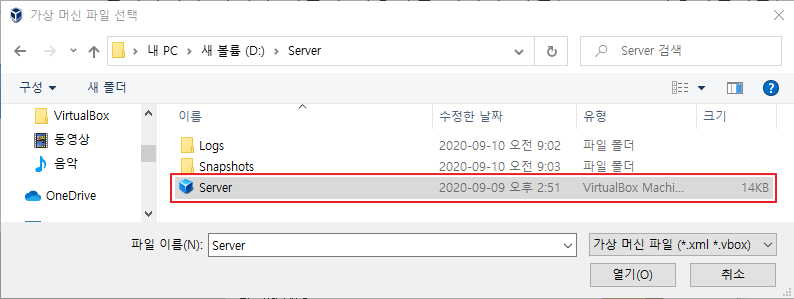


* 모든 파일 지우기 : VirtualBox 관리자에서 삭제와 더불어 가상머신 이미지 파일(C:\Linux\가상머신이름)도 함께 삭제 ⇒ 더 이상 사용하지 않는 가상머신을 삭제해서 디스크 공간을 확보하기 위한 용도
* **지우기만 하기** : VirtualBox 관리자에서만 삭제되고 가상머신 이미지 파일은 그대로 보존

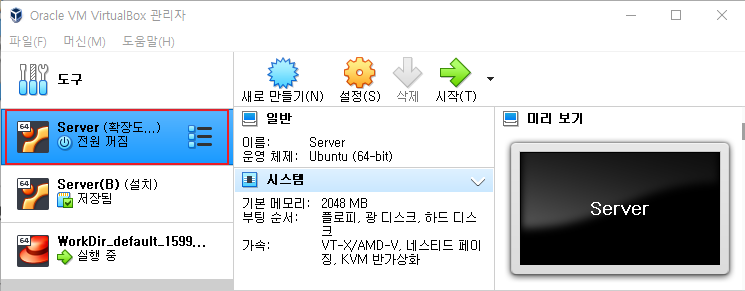
1. 가상머신 이미지 파일이 저장된 폴더를 다른 곳으로 이동
2. VirtualBox 관리자에서 추가 버튼을 클릭해서 이동한 폴더의 VirualBox 이미지를 선택



* 가져오기 : **ovf, ova 형식(가상머신 프로그램 간 상호호환 가능한 포맷)**의 파일을 이용해서 가상머신을 추가
* 내보내기 : ovf, ova 형식으로 가상머신 파일을 생성
* **추가** : VirtualBox에서 만들어진 가상머신 이미지를 추가



1. 가상머신 추가를 확인



# 오늘 숙제 : OVA 또는 OVF 파일로 만들고 다시 복원 (내보내기, 가져오기)

# vagrant 로컬 레포지터리 위치

C:\Users\myanj\.vagrant.d\boxes

~~~~~~~~~~~~~~ 현재 윈도우 사용자의 홈 디렉터리

# vagrant up 했을 때 생성되는 이미지 파일은 VirtualBox에서 사용하는 기본 저장소에 생성

C:\Users\myanj\VirtualBox VMs

# 가상 머신 정지하지고 삭제

C:\HashiCorp\WorkDir> vagrant halt

==> default: Attempting graceful shutdown of VM...

C:\HashiCorp\WorkDir> vagrant destroy

default: Are you sure you want to destroy the 'default' VM? [y/N] y

==> default: Destroying VM and associated drives...

# 가상 머신 생성 및 실행

C:\HashiCorp\WorkDir> vagrant up

# Vagrantfile로 인프라를 구성했을 때 장점

* 환경 구축 작업이 간소
* 환경 공유 용이
* 환경 파악 용이
* 팀 차원의 유지보수 가능

잠시 쉬고, 10시 15분에 이어서 진행하겠습니다.

**참고**

**언플러그드 컴퓨팅** ⇒ http://computing.or.kr/pds/CSUnpluggedv3.1KoreanVersion.pdf

# 보다 효율적으로 인프라를 관리하기 위한 개선 사항 (= vagrant의 한계)

구축 절차를 기술하는 사람에 따라 다양

구축된 환경에 대한 추가 설정이 불가

다양한 환경에 적용하기 어렵다

# 인프라 구성 관리 도구

선언적

추상화

수렴화

멱등성

간소화

# 앤서블(Ansible) 기본 사용법

파이썬으로 만들어진 인프라 구성 관리 도구

앤서블 본체

인벤터리(inventory)

모듈(module)

환경 설정 및 구축 절차를 통일된 형식으로 기술

매개 변수 등 환경의 차이를 관리

실행 전에 변경되는 부분을 파악

### #1 nginx 설치되어 있는지 확인

C:\HashiCorp\WorkDir> vagrant ssh

[vagrant@demo ~]$ systemctl status nginx

● nginx.service - The nginx HTTP and reverse proxy server

Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)

Active: active (running) since Thu 2020-09-10 00:38:06 UTC; 53min ago ⇐ 설치되어 실행 중

Process: 3217 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)

Process: 3214 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)

Process: 3212 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)

Main PID: 3219 (nginx)

CGroup: /system.slice/nginx.service

├─3219 nginx: master process /usr/sbin/nginx

├─3220 nginx: worker process

└─3221 nginx: worker process

[vagrant@demo ~]$

### #2 ansible 설치 및 버전 확인

[vagrant@demo ~]$ sudo systemctl stop nginx.service

[vagrant@demo ~]$ sudo yum install -y epel-release

[vagrant@demo ~]$ sudo yum install -y ansible

[vagrant@demo ~]$ ansible --version

ansible 2.9.10

config file = /etc/ansible/ansible.cfg

configured module search path = [u'/home/vagrant/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python2.7/site-packages/ansible

executable location = /usr/bin/ansible

python version = 2.7.5 (default, Apr 2 2020, 13:16:51) [GCC 4.8.5 20150623 (Red Hat 4.8.5-39)]

### #3 ansible 명령으로 nginx를 기동

[vagrant@demo ~]$ sudo sh -c "echo \"localhost\" >> /etc/ansible/hosts" ⇐ 서버 인번터리에 localhost를 추가

[vagrant@demo ~]$ cat /etc/ansible/hosts

# This is the default ansible 'hosts' file.

:

localhost

[vagrant@demo ~]$ ansible localhost -b -c local -m service -a "name=nginx state=started"

~~~~~~~~~ ~~~ ~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

| | | | --> 모듈의 추가 인자

| | | +--> service 모듈을 이용

| | +--> 대상 서버가 자기 자신이므로 SSH를 사용하지 않고 local로 연결

| +--> 원격 실행되는 대상 서버의 사용자 (-b = root)

+--> 인벤터리 파일에 기재된 서버 중 이번 명령어를 수행할 대상

localhost | CHANGED => {

"ansible\_facts": {

"discovered\_interpreter\_python": "/usr/bin/python"

},

"changed": true, ⇐ 중지 상태에서 실행 상태로 변경

"name": "nginx",

"state": "started",

"status": {

:

}

}

[vagrant@demo ~]$

### #4 nginx 상태 확인

[vagrant@demo ~]$ systemctl status nginx.service

● nginx.service - The nginx HTTP and reverse proxy server

Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)

Active: active (running) since Thu 2020-09-10 01:58:15 UTC; 7min ago

Process: 25830 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)

Process: 25828 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)

Process: 25826 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)

Main PID: 25832 (nginx)

CGroup: /system.slice/nginx.service

├─25832 nginx: master process /usr/sbin/nginx

├─25833 nginx: worker process

└─25834 nginx: worker process

### #5 nginx가 실행 상태일 때 ansible 명령(nginx 서비스를 시작)를 실행

[vagrant@demo ~]$ ansible localhost -b -c local -m service -a "name=nginx state=started"

localhost | SUCCESS => {

"ansible\_facts": {

"discovered\_interpreter\_python": "/usr/bin/python"

},

"changed": false, ⇐ 실행 상태였기 때문에, 상태 변화가 발생하지 않았음

"name": "nginx",

"state": "started",

"status": {

:

}

}

[vagrant@demo ~]$

잠시 쉬고 11시 25분에 이어서 진행하겠습니다.

# ansible-playbook (P78)

### #1 git 설치

[vagrant@demo ~]$ sudo yum install -y git

### #2 ansible-playbook-sample 레포지터리 클론 생성

[vagrant@demo ~]$ git clone https://github.com/devops-book/ansible-playbook-sample.git

### #3 playbook을 실행해서 구축

[vagrant@demo ~]$ cd ansible-playbook-sample/

[vagrant@demo ansible-playbook-sample]$ ansible-playbook -i development site.yml

~~~~~~~~~~~~~~

인벤터리 파일을 지정(/etc/ansible/hosts를 사용하지 않음)

[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv

to see details

PLAY [webservers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [common : install epel] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost] ⇐ epel 패키지가 이미 설치되어 있어 아무것도 하지 않았다.

TASK [install nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [nginx : replace index.html] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [localhost] ⇐ TASK 실행 결과가 예상했던 데로 변경되었다.

TASK [nginx start] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [localhost]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

localhost : ok=5 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[vagrant@demo ansible-playbook-sample]$

[vagrant@demo ansible-playbook-sample]$ curl localhost

hello, development ansible ⇐ nginx 홈 디렉터리의 index.html 파일의 내용

[vagrant@demo ansible-playbook-sample]$ ansible-playbook -i production site.yml

[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv

to see details

PLAY [webservers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [common : install epel] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [install nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [nginx : replace index.html] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [localhost]

TASK [nginx start] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

localhost : ok=5 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[vagrant@demo ansible-playbook-sample]$ curl localhost

hello, production ansible

[vagrant@demo ansible-playbook-sample]$ vi site.yml

|  |
| --- |
| --- ⇐ YAML 문서를 의미  - hosts: webservers ⇐ 배포 대상 호스트 ----------+ 하나의 플레이어를 정의  become: yes ⇐ 대상 호스트에 root 사용자로 작업을 수행하도록 지정 |  connection: local ⇐ 대상 호스트가 원격이 아니므로 ssh 대신 local로 연결 |  roles: ⇐ 롤 중 common과 nginx 롤을 실행 |  - common |  - nginx |  # - serverspec |  # - serverspec\_sample |  # - jenkins ----------+ |

점심 식사하고, 13시 10분에 이어서 진행하겠습니다.

### #4 실행 대상 정의를 확인

인벤터리 파일은 /etc/ansible/hosts를 사용하지만 -i 옵션을 이용해서 지정할 수 도 있음

[vagrant@demo ansible-playbook-sample]$ ls

development group\_vars production roles site.yml ⇐ 인벤터리 파일

~~~~~~~~~~~ ~~~~~~~~~~

[vagrant@demo ansible-playbook-sample]$ cat development

[development-webservers] ⇐ 그룹

localhost

[webservers:children]

development-webservers

[vagrant@demo ansible-playbook-sample]$ cat production

[production-webservers]

localhost

[webservers:children]

production-webservers

### #5 실행 내용 정의를 확인

ansible-playbook-sample/roles 디렉터리 아래에 tasks 디렉터리에 기록

[vagrant@demo ansible-playbook-sample]$ ls ./roles

common jenkins nginx serverspec serverspec\_sample ⇐ 롤 별로 실행될 내용을 담고 있는 디렉터리

[vagrant@demo ansible-playbook-sample]$ ls ./roles/common/tasks/

main.yml ⇐ common 롤에서 수행해야 할 내용을 정의

[vagrant@demo ansible-playbook-sample]$ ls ./roles/nginx/tasks/

main.yml ⇐ nginx 롤에서 수행해야 할 내용을 정의

[vagrant@demo ansible-playbook-sample]$ cat ./roles/common/tasks/main.yml

---

# tasks file for common

- name: install epel

yum: name=epel-release state=installed

[vagrant@demo ansible-playbook-sample]$ cat ./roles/nginx/tasks/main.yml

---

# tasks file for nginx

- name: install nginx

yum: name=nginx state=installed

- name: replace index.html

template: src=index.html.j2 dest=/usr/share/nginx/html/index.html

- name: nginx start

service: name=nginx state=started enabled=yes

### #6 템플릿 확인

[vagrant@demo ansible-playbook-sample]$ ls ./roles/nginx/templates/

index.html.j2

[vagrant@demo ansible-playbook-sample]$ cat ./roles/nginx/templates/index.html.j2

hello, {{ env }} ansible

### #7 템플릿에서 사용하는 변수 값을 확인

템플릿에서 사용하는 변수 값은 group\_vars 디렉터리에 그룹별로 정의

[vagrant@demo ansible-playbook-sample]$ ls ./group\_vars/

development-webservers.yml production-webservers.yml

[vagrant@demo ansible-playbook-sample]$ cat ./group\_vars/development-webservers.yml

env: "development"

[vagrant@demo ansible-playbook-sample]$ cat ./group\_vars/production-webservers.yml

env: "production"

### #8 템플릿 내용을 변경

[vagrant@demo ansible-playbook-sample]$ vi ./roles/nginx/templates/index.html.j2

HELLO, {{ env }} ansible !!! ⇐ 템플릿 내용 변경

### #9 dry-run 모드로 실행

실제로 변경된 내용을 반영하지 않고 반영되었을 때 결과를 미리 확인

[vagrant@demo ansible-playbook-sample]$ ansible-playbook -i development site.yml --check --diff

~~~~~~~ ~~~~~~

| |

| +-- 변경 차이를 표시

+-- dry-run 모드로 실행

[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [webservers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [common : install epel] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [install nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [nginx : replace index.html] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--- before: /usr/share/nginx/html/index.html

+++ after: /home/vagrant/.ansible/tmp/ansible-local-266825H2C\_L/tmpOtW5Mb/index.html.j2

@@ -1 +1 @@

-hello, production ansible

+HELLO, development ansible !!!

changed: [localhost]

TASK [nginx start] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

localhost : ok=5 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[vagrant@demo ansible-playbook-sample]$ curl localhost

hello, production ansible ⇐ **run-dry 모드**로 실행했기 때문에 이전과 동일한 상태로 실행

### #10 변경 사항을 호스트에 반영

[vagrant@demo ansible-playbook-sample]$ ansible-playbook -i development site.yml --diff

[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [webservers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [common : install epel] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [install nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [nginx : replace index.html] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--- before: /usr/share/nginx/html/index.html

+++ after: /home/vagrant/.ansible/tmp/ansible-local-26883nYwnzL/tmpOTC0eB/index.html.j2

@@ -1 +1 @@

-hello, production ansible

+HELLO, development ansible !!!

changed: [localhost]

TASK [nginx start] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

localhost : ok=5 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[vagrant@demo ansible-playbook-sample]$ curl localhost

HELLO, development ansible !!! ⇐ 변경된 내용이 반영된 것을 확인

# Vagrant와 Ansible의 차이에 대해 생각해 보세요

# What is DevOps ?

⇒ https://www.youtube.com/watch?time\_continue=1&v=\_I94-tJlovg&feature=emb\_logo

잠시 쉬고 14시 25분에 이어서 진행하겠습니다.

# 인프라 테스트 자동화 (P90)

## Serverspec

* 테스트 수행을 간단하고 쉽게 하기 위한 도구
* 인프라(서버) 설정 테스트 가능
* 테스트 항목에 대한 목록을 정해진 포맷을 기반으로 기술이 가능
* 테스트 결과를 리포트 형식으로 출력이 가능

## Ansible을 이용해서 Serverspec을 설치

### #1 rvm 및 ruby 설치

[vagrant@demo ansible-playbook-sample]$ command curl -sSL https://rvm.io/mpapis.asc | sudo gpg2 --import -

gpg: directory `/root/.gnupg' created

gpg: new configuration file `/root/.gnupg/gpg.conf' created

gpg: WARNING: options in `/root/.gnupg/gpg.conf' are not yet active during this run

gpg: keyring `/root/.gnupg/secring.gpg' created

gpg: keyring `/root/.gnupg/pubring.gpg' created

gpg: /root/.gnupg/trustdb.gpg: trustdb created

gpg: key D39DC0E3: public key "Michal Papis (RVM signing) <mpapis@gmail.com>" imported

gpg: Total number processed: 1

gpg: imported: 1 (RSA: 1)

gpg: no ultimately trusted keys found

[vagrant@demo ansible-playbook-sample]$ command curl -sSL https://rvm.io/pkuczynski.asc | sudo gpg2

--import -

gpg: key 39499BDB: public key "Piotr Kuczynski <piotr.kuczynski@gmail.com>" imported

gpg: Total number processed: 1

gpg: imported: 1 (RSA: 1)

[vagrant@demo ansible-playbook-sample]$ curl -L get.rvm.io | sudo bash -s stable

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 194 100 194 0 0 340 0 --:--:-- --:--:-- --:--:-- 340

100 24535 100 24535 0 0 10527 0 0:00:02 0:00:02 --:--:-- 16488

Downloading https://github.com/rvm/rvm/archive/1.29.10.tar.gz

Downloading https://github.com/rvm/rvm/releases/download/1.29.10/1.29.10.tar.gz.asc

gpg: Signature made Wed 25 Mar 2020 09:58:42 PM UTC using RSA key ID 39499BDB

gpg: Good signature from "Piotr Kuczynski <piotr.kuczynski@gmail.com>"

gpg: WARNING: This key is not certified with a trusted signature!

gpg: There is no indication that the signature belongs to the owner.

Primary key fingerprint: 7D2B AF1C F37B 13E2 069D 6956 105B D0E7 3949 9BDB

GPG verified '/usr/local/rvm/archives/rvm-1.29.10.tgz'

Creating group 'rvm'

Installing RVM to /usr/local/rvm/

Installation of RVM in /usr/local/rvm/ is almost complete:

\* First you need to add all users that will be using rvm to 'rvm' group,

and logout - login again, anyone using rvm will be operating with `umask u=rwx,g=rwx,o=rx`.

\* To start using RVM you need to run `source /etc/profile.d/rvm.sh`

in all your open shell windows, in rare cases you need to reopen all shell windows.

\* Please do NOT forget to add your users to the rvm group.

The installer no longer auto-adds root or users to the rvm group. Admins must do this.

Also, please note that group memberships are ONLY evaluated at login time.

This means that users must log out then back in before group membership takes effect!

Thanks for installing RVM 🙏

Please consider donating to our open collective to help us maintain RVM.

👉 Donate: https://opencollective.com/rvm/donate

[vagrant@demo ansible-playbook-sample]$ sudo usermod -aG rvm $USER

[vagrant@demo ansible-playbook-sample]$ id $USER

uid=1000(vagrant) gid=1000(vagrant) groups=1000(vagrant),1001(rvm)

[vagrant@demo ansible-playbook-sample]$ source /etc/profile.d/rvm.sh

[vagrant@demo ansible-playbook-sample]$ rvm reload

RVM reloaded!

[vagrant@demo ansible-playbook-sample]$ sudo su

[root@demo ansible-playbook-sample]# rvm requirements run

Checking requirements for centos.

Installing requirements for centos.

Installing required packages: bison, libffi-devel, readline-devel, sqlite-devel, zlib-devel, openssl-devel............

Requirements installation successful.

[root@demo ansible-playbook-sample]# rvm install 2.7

Searching for binary rubies, this might take some time.

Found remote file https://rvm\_io.global.ssl.fastly.net/binaries/centos/7/x86\_64/ruby-2.7.0.tar.bz2

Checking requirements for centos.

Requirements installation successful.

ruby-2.7.0 - #configure

ruby-2.7.0 - #download

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 18.3M 100 18.3M 0 0 73231 0 0:04:22 0:04:22 --:--:-- 74837

No checksum for downloaded archive, recording checksum in user configuration.

ruby-2.7.0 - #validate archive

ruby-2.7.0 - #extract

ruby-2.7.0 - #validate binary

ruby-2.7.0 - #setup

ruby-2.7.0 - #gemset created /usr/local/rvm/gems/ruby-2.7.0@global

ruby-2.7.0 - #importing gemset /usr/local/rvm/gemsets/global.gems..................................

ruby-2.7.0 - #generating global wrappers.......

ruby-2.7.0 - #gemset created /usr/local/rvm/gems/ruby-2.7.0

ruby-2.7.0 - #importing gemsetfile /usr/local/rvm/gemsets/default.gems evaluated to empty gem list

ruby-2.7.0 - #generating default wrappers.......

[root@demo ansible-playbook-sample]# rvm use 2.7 --default

Using /usr/local/rvm/gems/ruby-2.7.0

[root@demo ansible-playbook-sample]# rvm list

=\* ruby-2.7.0 [ x86\_64 ]

# => - current

# =\* - current && default

# \* - default

[root@demo ansible-playbook-sample]# exit

[vagrant@demo ansible-playbook-sample]$ ruby -v

ruby 2.0.0p648 (2015-12-16) [x86\_64-linux]

[vagrant@demo ansible-playbook-sample]$ sudo ruby -v

ruby 2.0.0p648 (2015-12-16) [x86\_64-linux]

[vagrant@demo ansible-playbook-sample]$ rvm use 2.7 --default

Using /usr/local/rvm/gems/ruby-2.7.0

[vagrant@demo ansible-playbook-sample]$ ruby -v

ruby 2.7.0p0 (2019-12-25 revision 647ee6f091) [x86\_64-linux]

[vagrant@demo ansible-playbook-sample]$ sudo ruby -v

ruby 2.0.0p648 (2015-12-16) [x86\_64-linux]

[vagrant@demo ansible-playbook-sample]$ which ruby

/usr/local/rvm/rubies/ruby-2.7.0/bin/ruby

[vagrant@demo ansible-playbook-sample]$ sudo which ruby

/bin/ruby

[vagrant@demo ansible-playbook-sample]$ sudo mv /bin/ruby /bin/ruby\_2.0.0

[vagrant@demo ansible-playbook-sample]$ sudo ln -s /usr/local/rvm/rubies/ruby-2.7.0/bin/ruby /bin/ruby

[vagrant@demo ansible-playbook-sample]$ ruby -v

ruby 2.7.0p0 (2019-12-25 revision 647ee6f091) [x86\_64-linux]

[vagrant@demo ansible-playbook-sample]$ sudo ruby -v

ruby 2.7.0p0 (2019-12-25 revision 647ee6f091) [x86\_64-linux]

### #2 Playbook 파일(site.yml)에서 serverspec 롤을 추가

[vagrant@demo ansible-playbook-sample]$ vi site.yml

|  |
| --- |
| ---  - hosts: webservers  become: yes  connection: local  roles:  - common  - nginx  - serverspec ⇐ 주석(#) 해제 후 저장  # - serverspec\_sample  # - jenkins |

### #3 Serverspec 롤을 확인

[vagrant@demo ansible-playbook-sample]$ cat ./roles/serverspec/tasks/main.yml

---

# tasks file for serverspec

- name: install ruby

yum: name=ruby state=installed

- name: install serverspec

gem: name={{ item }} state=present user\_install=no

with\_items:

- rake

- serverspec

### #4 ansible-playbook으로 Serverspec 설치

[vagrant@demo ansible-playbook-sample]$ ansible-playbook -i development site.yml --diff

[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [webservers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [common : install epel] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [install nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [nginx : replace index.html] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [nginx start] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [serverspec : install ruby] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [install serverspec] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost] => (item=rake)

changed: [localhost] => (item=serverspec)

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

localhost : ok=7 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

잠시 쉬고 15시 40분에 이어서 진행하겠습니다.

### #5 Serverspec 설정

[vagrant@demo ansible-playbook-sample]$ serverspec-init

Select OS type:

1) UN\*X

2) Windows

Select number: 1

Select a backend type:

1) SSH

2) Exec (local)

Select number: 2

+ spec/

+ spec/localhost/

+ spec/localhost/sample\_spec.rb ⇐ 샘플

+ spec/spec\_helper.rb

+ Rakefile

+ .rspec

### #6 sample\_spec.rb 파일을 확인 ⇒ 테스트 케이스 작성법을 확인

[vagrant@demo ansible-playbook-sample]$ cat ./spec/localhost/sample\_spec.rb ⇐ 서버의 상태를 테스트하는 코드

require 'spec\_helper'

describe package('httpd'), :if => os[:family] == 'redhat' do

it { should be\_installed } ⇐ httpd가 설치되어 있어야 한다.

end

:

describe service('httpd'), :if => os[:family] == 'redhat' do

it { should be\_enabled } ⇐ httpd 서비스가 활성화되어 있어야 한다.

it { should be\_running } ⇐ httpd 서비스가 실행되고 있어야 한다.

end

:

describe port(80) do

it { should be\_listening } ⇐ 80 포트가 동작(listen)하고 있어야 한다.

end

### #7 Serverspec을 이용한 테스트 실행

[vagrant@demo ansible-playbook-sample]$ rake spec

/usr/local/rvm/rubies/ruby-2.7.0/bin/ruby -I/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-support-3.9.3/lib:/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/lib /usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/exe/rspec --pattern spec/localhost/\\*\_spec.rb

Package "httpd"

is expected to be installed (FAILED - 1)

Service "httpd"

is expected to be enabled (FAILED - 2)

is expected to be running (FAILED - 3)

Port "80"

is expected to be listening

Failures: ⇐ 실패한 테스트 케이스에 대한 상세 설명

1) Package "httpd" is expected to be installed

On host `localhost'

Failure/Error: it { should be\_installed }

expected Package "httpd" to be installed

/bin/sh -c rpm\ -q\ httpd

package httpd is not installed

# ./spec/localhost/sample\_spec.rb:4:in `block (2 levels) in <top (required)>'

2) Service "httpd" is expected to be enabled

On host `localhost'

Failure/Error: it { should be\_enabled }

expected Service "httpd" to be enabled

/bin/sh -c systemctl\ --quiet\ is-enabled\ httpd

# ./spec/localhost/sample\_spec.rb:12:in `block (2 levels) in <top (required)>'

3) Service "httpd" is expected to be running

On host `localhost'

Failure/Error: it { should be\_running }

expected Service "httpd" to be running

/bin/sh -c systemctl\ is-active\ httpd

unknown

# ./spec/localhost/sample\_spec.rb:13:in `block (2 levels) in <top (required)>'

Finished in 0.08143 seconds (files took 0.45331 seconds to load)

4 examples, 3 failures ⇐ 테스트 결과 요약

Failed examples: ⇐ 실패한 테스트 케이스 목록

rspec ./spec/localhost/sample\_spec.rb:4 # Package "httpd" is expected to be installed

rspec ./spec/localhost/sample\_spec.rb:12 # Service "httpd" is expected to be enabled

rspec ./spec/localhost/sample\_spec.rb:13 # Service "httpd" is expected to be running

/usr/local/rvm/rubies/ruby-2.7.0/bin/ruby -I/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-support-3.9.3/lib:/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/lib /usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/exe/rspec --pattern spec/localhost/\\*\_spec.rb failed

## Ansible을 이용해서 Serverpec에서 사용하는 테스트 케이스(\_sepc.rb)를 자동으로 생성

### #1 Playbook 파일(site.yml)에 serverspec\_sample 롤(role)을 추가

[vagrant@demo ansible-playbook-sample]$ vi site.yml

|  |
| --- |
| ---  - hosts: webservers  become: yes  connection: local  roles:  - common  - nginx  - serverspec  - serverspec\_sample ⇐ 주석(#) 해제 후 저장  # - jenkins |

### #2 serverspec\_sample 롤 정의 파일을 확인

[vagrant@demo ansible-playbook-sample]$ cat ./roles/serverspec\_sample/tasks/main.yml

---

# tasks file for serverspec\_sample

- name: distribute serverspec suite

copy: src=serverspec\_sample dest={{ serverspec\_base\_path }} ⇐ /tmp 아래로 serverspec\_sample 디렉터리를 복사

- name: distribute spec file

template: src=web\_spec.rb.j2 dest={{ serverspec\_path }}/spec/localhost/web\_spec.rb

⇐ 템플릿에 정의된 내용으로 web\_spec.rb 파일을 생성

[vagrant@demo ansible-playbook-sample]$ cat ./roles/serverspec\_sample/vars/main.yml

serverspec\_base\_path: "/tmp" ⇐ task에서 사용하는 변수를 정의

serverspec\_path: "{{ serverspec\_base\_path }}/serverspec\_sample"

[vagrant@demo ansible-playbook-sample]$ cat ./roles/serverspec\_sample/templates/web\_spec.rb.j2

require 'spec\_helper' ⇐ serverspec에서 사용할 테스트 케이스 템플릿

describe package('nginx') do ⇐ nginx 설치 여부

it { should be\_installed }

end

describe service('nginx') do ⇐ nginx 실행/활성화 여부

it { should be\_enabled }

it { should be\_running }

end

describe port(80) do ⇐ 80 포트 확인

it { should be\_listening }

end

describe file('/usr/share/nginx/html/index.html') do

it { should be\_file } ⇐ index.html 파일 존재 여부

it { should exist }

its(:content) { should match /^Hello, {{ env }} ansible!!$/ } ⇐ index.html 파일의 내용 검증

end

### #3 ansible-playbook으로 spec 파일(테스트 케이스를 정의하고 있는 파일)을 배포

[vagrant@demo ansible-playbook-sample]$ ansible-playbook -i development site.yml

[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [webservers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [common : install epel] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [install nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [nginx : replace index.html] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [nginx start] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [serverspec : install ruby] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [install serverspec] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost] => (item=rake)

ok: [localhost] => (item=serverspec)

TASK [serverspec\_sample : distribute serverspec suite] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [localhost] ⇐ /tmp 디렉터리 아래로 serverspec\_sample 디렉터리를 복사

/tmp/serverspec\_sample 디렉터리는 인프라가 원하는 형태로 구성되었는지 테스트하는 공간

TASK [serverspec\_sample : distribute spec file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [localhost] ⇐ 템플릿을 이용해서 web\_spec.rb 파일을 정상적으로 생성

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

localhost : ok=9 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### #4 spec 파일(테스트 케이스를 정의) 생성을 확인

[vagrant@demo ansible-playbook-sample]$ cat /tmp/serverspec\_sample/spec/localhost/web\_spec.rb

require 'spec\_helper'

describe package('nginx') do

it { should be\_installed }

end

describe service('nginx') do

it { should be\_enabled }

it { should be\_running }

end

describe port(80) do

it { should be\_listening }

end

describe file('/usr/share/nginx/html/index.html') do

it { should be\_file }

it { should exist }

its(:content) { should match /^Hello, development ansible!!$/ }

end

### #5 (ansible을 이용해서 자동으로 생성한 spec 파일을 이용해서) 테스트를 실행

[vagrant@demo ansible-playbook-sample]$ cd /tmp/serverspec\_sample/ ⇐ 작업 디렉터리(테스트 디렉터리)로 이동

[vagrant@demo serverspec\_sample]$ rake spec ⇐ 테스트 실행

/usr/local/rvm/rubies/ruby-2.7.0/bin/ruby -I/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-support-3.9.3/lib:/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/lib /usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/exe/rspec --pattern spec/localhost/\\*\_spec.rb

Package "nginx"

is expected to be installed

Service "nginx"

is expected to be enabled

is expected to be running

Port "80"

is expected to be listening

File "/usr/share/nginx/html/index.html"

is expected to be file

is expected to exist

content

is expected to match /^Hello, development ansible!!$/ (FAILED - 1) ⇐ nginx의 index.html 파일 내용이 테스트 케이스에 명싱된 내용과

다르기 때문에 테스트 실패가 발생

Failures:

1) File "/usr/share/nginx/html/index.html" content is expected to match /^Hello, development ansible!!$/

On host `localhost'

Failure/Error: its(:content) { should match /^Hello, development ansible!!$/ }

expected "HELLO, development ansible !!!\n" to match /^Hello, development ansible!!$/

Diff:

@@ -1 +1 @@

-/^Hello, development ansible!!$/

+HELLO, development ansible !!!

/bin/sh -c cat\ /usr/share/nginx/html/index.html\ 2\>\ /dev/null\ \|\|\ echo\ -n

HELLO, development ansible !!!

# ./spec/localhost/web\_spec.rb:19:in `block (2 levels) in <top (required)>'

Finished in 0.11232 seconds (files took 0.40656 seconds to load)

7 examples, 1 failure

Failed examples:

rspec ./spec/localhost/web\_spec.rb:19 # File "/usr/share/nginx/html/index.html" content is expected to match /^Hello, development ansible!!$/

/usr/local/rvm/rubies/ruby-2.7.0/bin/ruby -I/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-support-3.9.3/lib:/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/lib /usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/exe/rspec --pattern spec/localhost/\\*\_spec.rb failed

### #6 테스트 케이스를 통과하도록 컨텐츠를 수정 → 컨텐츠 형식을 정의하고 있는 템플릿 파일을 수정

[vagrant@demo ansible-playbook-sample]$ cat ~/ansible-playbook-sample/roles/nginx/templates/index.html.j2

HELLO, {{ env }} ansible !!! ⇐ 테스트 케이스와 상이 → 테스트 케이스에 맞춰서 수정

[vagrant@demo ansible-playbook-sample]$ vi ~/ansible-playbook-sample/roles/nginx/templates/index.html.j2

Hello, {{ env }} ansible!!

### #7 ansible-playbook으로 수정한 템플릿에 맞춰서 새롭게 index.html을 생성

[vagrant@demo ansible-playbook-sample]$ ansible-playbook -i development site.yml

[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [webservers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [common : install epel] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [install nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [nginx : replace index.html] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [localhost]

TASK [nginx start] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [serverspec : install ruby] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [install serverspec] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost] => (item=rake)

ok: [localhost] => (item=serverspec)

TASK [serverspec\_sample : distribute serverspec suite] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

TASK [serverspec\_sample : distribute spec file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [localhost]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

localhost : ok=9 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### #8 테스트를 실행

[vagrant@demo ansible-playbook-sample]$ cd /tmp/serverspec\_sample/

[vagrant@demo serverspec\_sample]$ rake spec

/usr/local/rvm/rubies/ruby-2.7.0/bin/ruby -I/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-support-3.9.3/lib:/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/lib /usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/exe/rspec --pattern spec/localhost/\\*\_spec.rb

Package "nginx"

is expected to be installed

Service "nginx"

is expected to be enabled

is expected to be running

Port "80"

is expected to be listening

File "/usr/share/nginx/html/index.html"

is expected to be file

is expected to exist

content

is expected to match /^Hello, development ansible!!$/

Finished in 0.10557 seconds (files took 0.41014 seconds to load)

7 examples, 0 failures ⇐ 7개 테스트 케이스를 모두 통과

### #9 nginx를 중지 후 테스트 실행

[vagrant@demo serverspec\_sample]$ sudo systemctl stop nginx.service

[vagrant@demo serverspec\_sample]$ systemctl status nginx.service

● nginx.service - The nginx HTTP and reverse proxy server

Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)

Active: inactive (dead) since Thu 2020-09-10 08:12:46 UTC; 21s ago

Main PID: 25832 (code=exited, status=0/SUCCESS)

[vagrant@demo serverspec\_sample]$ rake spec

/usr/local/rvm/rubies/ruby-2.7.0/bin/ruby -I/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-support-3.9.3/lib:/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/lib /usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/exe/rspec --pattern spec/localhost/\\*\_spec.rb

Package "nginx"

is expected to be installed

Service "nginx"

is expected to be enabled

is expected to be running (FAILED - 1)

Port "80"

is expected to be listening (FAILED - 2)

File "/usr/share/nginx/html/index.html"

is expected to be file

is expected to exist

content

is expected to match /^Hello, development ansible!!$/

Failures:

1) Service "nginx" is expected to be running

On host `localhost'

Failure/Error: it { should be\_running }

expected Service "nginx" to be running

/bin/sh -c systemctl\ is-active\ nginx

inactive

# ./spec/localhost/web\_spec.rb:9:in `block (2 levels) in <top (required)>'

2) Port "80" is expected to be listening

On host `localhost'

Failure/Error: it { should be\_listening }

expected Port "80" to be listening

/bin/sh -c ss\ -tunl\ \|\ grep\ -E\ --\ :80\\\

# ./spec/localhost/web\_spec.rb:13:in `block (2 levels) in <top (required)>'

Finished in 0.11722 seconds (files took 0.41031 seconds to load)

7 examples, 2 failures

Failed examples:

rspec ./spec/localhost/web\_spec.rb:9 # Service "nginx" is expected to be running

rspec ./spec/localhost/web\_spec.rb:13 # Port "80" is expected to be listening

/usr/local/rvm/rubies/ruby-2.7.0/bin/ruby -I/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-support-3.9.3/lib:/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/lib /usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/exe/rspec --pattern spec/localhost/\\*\_spec.rb failed

[vagrant@demo serverspec\_sample]$

### #10 테스트 결과를 HTML 형식으로 출력

[vagrant@demo serverspec\_sample]$ sudo gem install coderay

[vagrant@demo serverspec\_sample]$ rake spec SPEC\_OPTS="--format html" > ~/result.html

/usr/local/rvm/rubies/ruby-2.7.0/bin/ruby -I/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-support-3.9.3/lib:/usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/lib /usr/local/rvm/rubies/ruby-2.7.0/lib/ruby/gems/2.7.0/gems/rspec-core-3.9.2/exe/rspec --pattern spec/localhost/\\*\_spec.rb failed

[vagrant@demo serverspec\_sample]$ sudo mv ~/result.html /usr/share/nginx/html/

[vagrant@demo serverspec\_sample]$ sudo setenforce 0

[vagrant@demo serverspec\_sample]$ sudo systemctl start nginx.service

호스트 PC에서 http://192.168.33.10/result.html 으로 접속

가상 머신으로 접속이 안 될 때 sudo systemctl stop firewalld 명령으로 방화벽 해제



의미 문자 = 메타 문자 ⇒ 어떤 기능에서 특별한 의미를 가지는 문자 = 특수 기호

예를 들면,

SQL 문을 작성할 때 홑따움표(')는 문자열의 시작과 끝을 나타내는 의미를 가진 문자

URL에서 &는 요청 파라미터와 파라미터를 구분하는 의미를 가진 문자

의미 문자를 문자 그 자체로 사용해야 할 경우가 있다.

예를 들면,

content 컬럼에 John's Name 이라는 글자가 들어간 것을 검색하는 쿼리를 만들 때 ⇒ content like '%John's Name%'

요청 파라미터 중 파라미터 이름이 CompanyName이고, 파라미터 값이 Bandi & Luce인 경우 ⇒ …?CompanyName=Bandi & Luce

⇒ 의미 문자를 그대로 사용하면 해석을 할 수 없거나 잘못 해석해서 오류가 발생

⇒ 이런 문제점을 해결하기 위해서는 의미 문자를 문자 그 자체로 해석될 수 있도록 변형해야 한다.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

이스케이프 ⇒ 의미 문자에서 의미를 제거하고 문자만 남기는 것

이스케이프를 구현하는 방법

1. **이스케이프를 나타내는 의미 문자를 사용 ⇒ \ (역슬래쉬)** ⇒ content like '%John\'s Name%'
2. 해당 기능에서 제공해 주는 규칙을 이용 ⇒ MySQL인 경우 홑따움표를 두 번 사용 ⇒ content like '%John''s Name%'
3. 일정한 규칙에 따라서 변경해서 이용 = 인코딩 ⇒ URL의 경우 URL Encoding 기법을 이용해서 표현 ⇒ CompanyName=Bandi %26 Luce

# Git, GitHub 부분은 자율 학습 (P104~P145)

내일은 3장 부터 진행 ...