Linux 명령어

**Ubuntu server Tip**

1. 가상호스팅을 받은 우분투 서버를 보면, 글자가 영어임에도 불구하고 다이아몬드로 깨지는 현상이 발생합니다. 이런 경우 export LANG=C라고 명령어를 준 다음부터는 잘 나오지만 부팅을 다시 하면 또 다시 깨지게 된다. 이럴 경우 nano ~/.bashrc 하여 가장 마지막 줄에 export LANG=C를 추가하고 저장하면 깨짐 현상을 해결 할 수 있다.

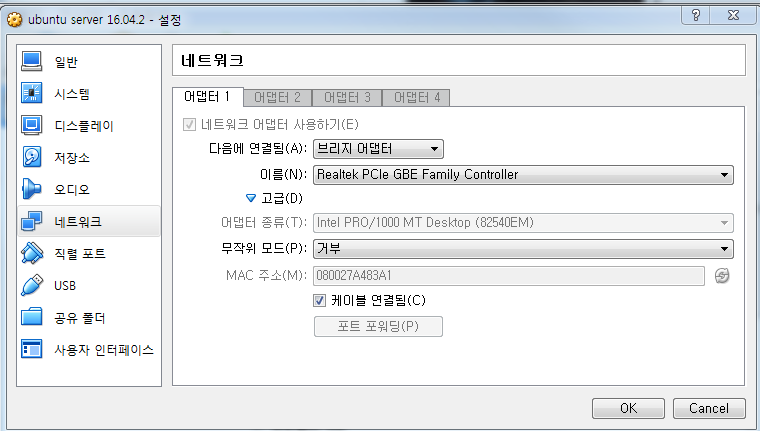
2. 콘솔에서 화면 scroll : shift + page up

3. Network방식을 변경하다 보면 network설정에서 많은 시간을 소요함 이럴 경우 network 아이콘 오른쪽 마우스버튼 네트워크 어댑터설정

4. putty를 사용하려면 VirtualBox에서 network은 bridge방식으로 선언되어야 하

며, 유선으로 Lan이 연결되도록 하여야 한다. 이렇게 될 경우 외부에서 우분투

서버에 접근 할 수 있을 것으로 판단됨.



5. /etc/network/interfaces의 내용 확인 할 것

1. # This file describes the network interfaces available on your system
2. # and how to activate them. For more information, see interfaces(5).
3. # The loopback network interface
4. auto lo
5. iface lo inet loopback
6. # The primary network interface
7. auto enp0s3
8. iface enp0s3 inet dhcp

https://www.youtube.com/watch?v=fCmSV2BE\_Sw

6. 무선공유기



Default gateway : 192.168.219.1

7.  [Virtual Box] : Host로부터 Guest로 고정 IP로 연결하기

/etc/network/interfaces

# The loopback network interface

auto lo

iface lo inet loopback

# The loopback network interface

auto enp0s3

#iface enp0s3 inet dhcp : original

iface enp0s3 inet static

address 192.168.219.109

netmask 255.255.255.0

gateway 192.168.219.1

network 192.168.219.0

broadcast 192.168.219.255

8.  sudo apt-get 실행시 error발생시

/etc/apt/sources.list에서

kr.archive.ubuntu.com 을 [ftp.daum.net](ftp://ftp.daum.net)으로 수정

/etc/resolv.conf를 수정

nameserver 168.126.63.1

9. root계정으로 로그인을 할 경우

1) sudo passwd root 명령어를 통해 root계정의 password 설정

2) su 명령어와 1)에서 설정한 password로 root권한 취득

10. root 계정으로 ssh 접속하기

root 계정으로 ssh 접속을 하기 위해서는 /etc/ssh/sshd\_config 파일을 수정해야 한다.

*# Authentication:*

*LoginGraceTime 120*

*PermitRootLogin prohibit-password*

*StrictModes yes*

*prohibit-password*를 *yes*로 바꾼뒤, 다음과 같이 명령하여 ssh를 다시 시작한다.

*service ssh restart*

11. 시스템 시작시

1) 부팅메시지 : /var/log/dmesg로 확인

2) Shell부여

- 기본적으로 bash shell

- root 사용자 : #, 일반 사용자는 $임

3) log out : 사용자가 빠져나감

컴퓨터끄기 : System다운

12. 가상콘솔

1) 기본적으로 6개까지 제공

2) Ctrl + Alt + f1 ~ f6까지 : 텍스트 모드 (tty1~tty6까지)

3) Ctrl + Alt + f7 : X윈도우

4) X윈도우 -> 콘솔 : Ctrl + Alt + backspace

5) X윈도우에서 터미날사용

.바탕화면>>오른쪽마우스 >> 터미날

**Linux 명령어**

1. mkdir : directory 만들기

2. ls -al : dir/w

ls -l

ls

3. touch :file 만들기

4. cd directory명 일부 + tab key :

5. rm : file 삭제

rm - r hello\_linux : directory삭제

rm -rf / : 묻지도 않고 root의 모든 것을 삭제함... 주의

6. 명령어 --help

rm --help

7. man 명령어 : Manual

8. mv : 화일을 리네임하거나 다를 곳으로 이동시킬 때 사용.

9. sudo : super관리자의 권한으로 명령을 실행하는 방법

super user do

Unix계열은 주로 다중사용자 용이었으며 따라서 각각의 사용자마다

permission(권한)을 부여 하였음.

apt-get install git : 권한이 없으면 실행을 할 수 없으나 sudo apt-get install

git을 할 수 있음.

10. Utility

1) grep : 해당 파일을 읽어서 해당 패턴을 검색하고, 패턴을 포함하는 줄의

내용을 출력한다.

grep xxx you.txt : you.txt에서 xxx가 포함된 Text를 출력

grep - : option

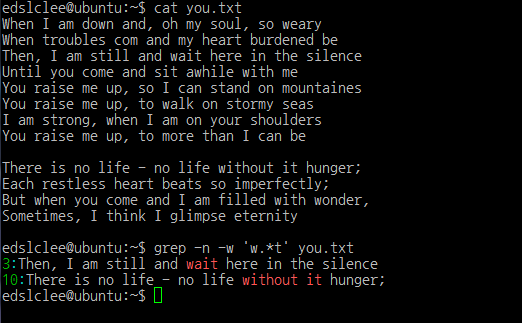
-i : 대소문자 구별 안 함.

-n: 라인 번호 포함

-w: 단어로 검색

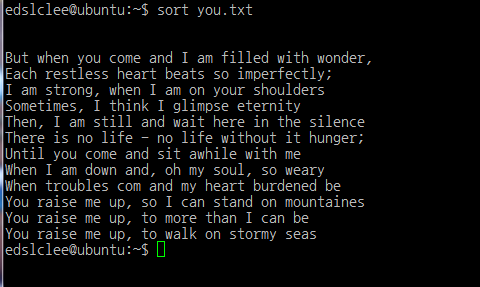
-v: reverse 조건으로 검색

-w ‘w.\*t’처럼 정규표현식으로 검색할 수 도 있음.



2) Sort : Text파일을 줄 단위로 정렬한다

- 기본 Format : sort [-옵션] 파일\*



- sort +2 -3 you.txt : 3번째 단어 기준으로 sort

3) split(파일 자르기) : 하나의 파일을 일정한 크기의 여러 개의 파일로 분할

-기본 format: split [-옵션] 입력파일 [출력파일]

- split –l 10 you.txt

ls -l

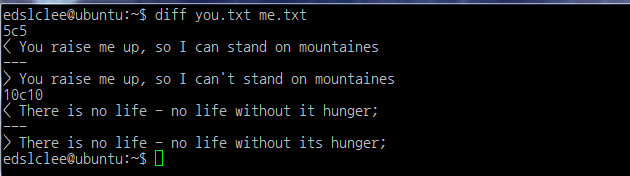


4) 파일비교

- cmp : cmp 파일1 파일2



- diff : diff 파일1 파일2



5) 파일찾기

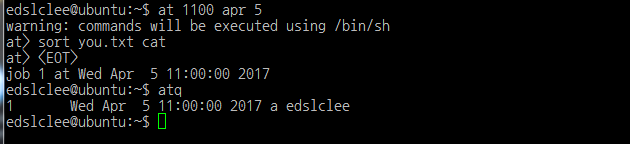
-format : find directory [-옵션]

6) 명령어 스케쥴링

- **$crontab 파일** : crontab파일을 cron시스템에 등록한다.

crontab –l : 등록된 crontab파일 list, -e: 수정 혹은 생성, -r: 삭제

- **$at [옵션] 시간**

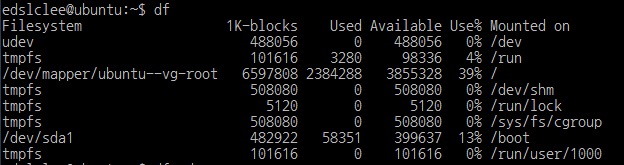
 $at –r jobnumber : 해당 at작업번호를 삭제

7) 디스크 및 아카이브

- **디스크 사용**

$ df 파일시스템\* : 파일명이 있을 경우 그 파일만, 파일명이 없을 경우 모든

파일 시스템에 대한 정보를 나타냄



$du [-s] 파일명 : 파일의 디스크 사용량을 나타냄

- **아카이브**

tar : 백업 또는 다른 장소로의 이동을 위해 파일을 묶어 놓은 묶음을 말하며,

tar(tape arcive)유틸리티를 사용하여 만들거나 푼다

$tar –[option] c(create), v(vervose, x(extract), t(table of contents), f(file)

UNIX에서는 압축을 할 경우 먼저 tar로 여러 파일을 묶은 후에 compress

또는 gzip으로 압축한다.

- **압축**

Compress : 유닉스에서 기본기본으로 제공하는 압축프로그램으로

. $compress 파일\*

. $uncompress 파일.Z

gzip: 윈도즈의 zip프로그램과 유사

. gzip 파일\*

. gzip –d 파일.gz

8) AWK

11. nano Editor사용법



^G: Get Help ^O: Write Out ^W :Where is ^K cut Text ^J: Justify ^C :Cur Pos ^Y: Prev Page

^X: Exit ^R: read file ^\:replace ^U un cut ^T: To spell

ctrl+^                         블럭지정하기 (흰색만 선택한 것임)    
alt-^     alt-shift-6      선택 종료후 컷버퍼에 복사, 선택 없으면 현재 라인을 복사  
^U                              붙여넣기  
^K                              잘라내기  
alt-R                           바꾸기  
alt-G                          지정한 라인, 컬럼으로 이동  
alt-W                         마지막 검색 반복  
^W                            원하는 문자열 찾기  
^G                             도움말   
^O                             새로운 파일 생성  
^X                              종료  
^R                              파일을 읽어오기  
alt-T                           커서부터 파일 끝까지 잘라내기  
alt-+                           화면을 줄 단위로 이동  
alt--                            화면을 줄 단위로 이동

12. Package Manager

LINUX의 package manager에는 apt와 yam이 있음.

1) sudo apt-get update; 로 update대상 catalog를 읽음

2) sudo apt-cache search htop

3) sudo apt-get install htop

1) ~ 3)의 순으로 package를 install.

4) sudo apt-get upgrade htop

5) sudo apt-get remove htop : 삭제

6) sudo apt-get purge apache : 더욱 강력함.

\* htop은 sudo htop을 하는 것이 좋다.

\* Bitnami LAMP는 /otc/lampstack~/uninstall

13. file을 다운로드 받을경우

1) wget을 사용

wget url/.\*

2) git을 활용

git clone https://www.github/edslclee/\*.git \* directory명

14. IO redirection \* http://slideplayer.com/slide/5117573/ 참고필요 (shell script)

1) stdout의 redirection

ls -l > result.text : ls -l의 출력 결과를 result.txt에

cat result.txt : 출력하기

- rm test.txt > log.txt명령어를 사용 할 경우 test.txt가 존재하지 않으면

screen상으로 error message가 출력됨. 이는 rm test.txt > log.txt는

rm test.txt 1 > log.txt이기 때문 즉 stdout임. file로 error message를 저장하려면

rm test.txt 2 > log.txt하면 stderr 형식이 되어 화일로 출력이 가능함.

따라서,

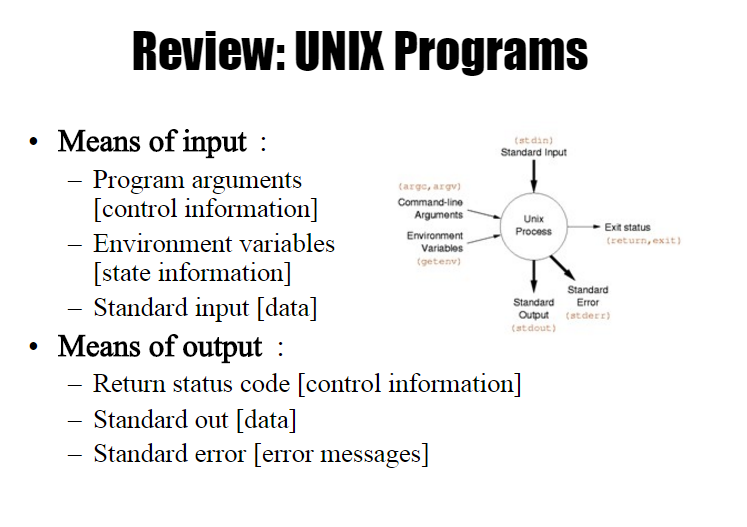
rm test.txt 1> result.txt 2>error.log 로 사용이 가능하다.

2) stdin의 redrection

cat < hello.txt : hello.txt의 내용을 입력으로 받는다.

head -n1 test.txt : test.text 1줄

head -n1 < test.txt > one.txt



3) etc

. ls -l >> result.txt : stdout redirection 결과를 Append

. mail edslclee@gmail.com << eot

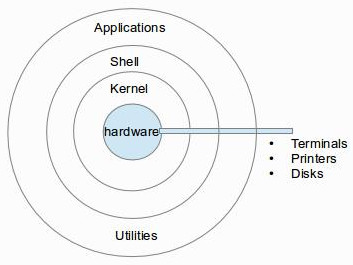
>hi

>djsjdsf

>eot

. ls -al > /dev/null : 쓰레기통으로

**15. Shell과 Kernel**



ls -l, ls;pwd등과 같이 입력한 명령을 shell이 받아서 kernel에게 전달. Kernel은

이 명령을 H/W에 전달 결과를 실행.

**16. bash vs zsh**

1) echo

echo $0 : bash

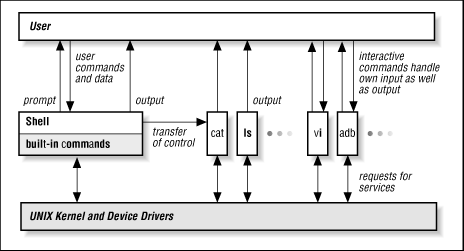
cd + tabkey : 숨긴 directory도 표시

2) zsh

echo $0 : zsh

cd + tabkey :

같은 kernel에 대해 다양한 shell이 존재할 수 있음.



ZSH에서 bash로 돌아 올 경우 : bash만 치면 됨.

**17. Shell Script**

여러 개의 shell명령어를 순서대로 실행할 수 있도록 하는 기능(Batch)

<script사용 전>

$mkdir script

$cd script

$touch a.log b.log c.log

$mkdir bak; cp \*.log bak

<Script사용>

$nano backup

#!/bin/bash : #! bin/bash를 통해서 실행된다는 것을 정의

if ! [ -d bak ]; then

mkdir bak

fi

cp \*.log bak

<실행을 하려면>

$./backup : permission denied

**$chmod +x backup : ls -l을 해보면 -rw-rw-r-- 1 backup이 -rwxrwxr-x로 바뀜 :권한 획득**

$./backup

<서버의 상태를 확인>

$ ~/script/serverstat

|  |
| --- |
| #!/bin/bash  sudo /opt/lampstack-5.6.30-1/ctlscript status |

**18. Directory**

google에서 linux directory structure 검색

http://www.thegeekstuff.com/2010/09/linux-file-system-structure/?utm\_source=tuicool

1. / – Root

* Every single file and directory starts from the root directory.
* Only root user has write privilege under this directory.
* Please note that /root is root user’s home directory, which is not same as /.

2. /bin – User Binaries 사용자가 사용하는 실행 프로그램들

* Contains binary executables.
* Common linux commands you need to use in single-user modes are located under this directory.
* Commands used by all the users of the system are located here.
* For example: ps, ls, ping, grep, cp.



3. /sbin – System Binaries

* Just like /bin, /sbin also contains binary executables.
* But, the linux commands located under this directory are used typically by system aministrator, for system maintenance purpose.
* For example: iptables, reboot, fdisk, ifconfig, swapon

4. /etc – Configuration Files

* Contains configuration files required by all programs.
* This also contains startup and shutdown shell scripts used to start/stop individual programs.
* For example: /etc/resolv.conf, /etc/logrotate.conf

5. /dev – Device Files

* Contains device files.
* These include terminal devices, usb, or any device attached to the system.
* For example: /dev/tty1, /dev/usbmon0

6. /proc – Process Information

* Contains information about system process.
* This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
* This is a virtual filesystem with text information about system resources. For example: /proc/uptime

7. /var – Variable Files

* var stands for variable files.
* Content of the files that are expected to grow can be found under this directory.
* This includes — system log files (/var/log); packages and database files (/var/lib); emails (/var/mail); print queues (/var/spool); lock files (/var/lock); temp files needed across reboots (/var/tmp);

8. /tmp – Temporary Files

* Directory that contains temporary files created by system and users.
* Files under this directory are deleted when system is rebooted.

9. /usr – User Programs

* Contains binaries, libraries, documentation, and source-code for second level programs.
* /usr/bin contains binary files for user programs. If you can’t find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
* /usr/sbin contains binary files for system administrators. If you can’t find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
* /usr/lib contains libraries for /usr/bin and /usr/sbin
* /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2

10. /home – Home Directories

* Home directories for all users to store their personal files.
* For example: /home/john, /home/nikita

11. /boot – Boot Loader Files

* Contains boot loader related files.
* Kernel initrd, vmlinux, grub files are located under /boot
* For example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic

12. /lib – System Libraries

* Contains library files that supports the binaries located under /bin and /sbin
* Library filenames are either ld\* or lib\*.so.\*
* For example: ld-2.11.1.so, libncurses.so.5.7

13. /opt – Optional add-on Applications

* opt stands for optional.
* Contains add-on applications from individual vendors.
* add-on applications should be installed under either /opt/ or /opt/ sub-directory.

14. /mnt – Mount Directory

* Temporary mount directory where sysadmins can mount filesystems.

15. /media – Removable Media Devices

* Temporary mount directory for removable devices.
* For examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer

16. /srv – Service Data

* srv stands for service.
* Contains server specific services related data.
* For example, /srv/cvs contains CVS related data.

**19. Process**

processor : process를 실행

Memory : Memory에 적재

Storage : program이 저장

Process Monitoring :

1) ps

ps -aux

ps aux | grep apache

sudo kill 21222

2) sudo top or htop

**20. file을 찾는법**

1) file의 위치를 찾는 방법

- $locate \*.log

- sudo updatedb

\* **http://www.tecmint.com/35-practical-examples-of-linux-find-command/**

- $find --help | head

- $find ~ -name \*.log : 본인의 directory 하위의 모든 directory를 검색

\*.log를 찾아라

$find / -name \*.log : 시스템 전체에서 찾아라 sudo를 붙여야 함.

2) 실행파일을 찾는 whereis, $PATH로 명령어가 어디에 위치하는가를 지정

- $whereis rm

- $echo $PATH : 환경변수

**21. Background 실행**

1) 두 개의 Program을 전환하는 방법

- $nano 로 edit하고 있다가 file을 save하되 종료하지 않고 ctr+Z를 누르면

다른 작업이 가능

- $fg를 누르면 foreground 인 nano로 전환

- $jobs : background에서 실행되는 file들을 볼 수 있음

- $vim : vi tool

- $Ctr-Z; $jobs : 2개의 shell nano와 vim이 stopped되어 있다

- $fg %2

-$kill %4 or kill -9 %4

2) 실행하면서 background로 작업을 수행

. ls -alR / > result.txt 2>error.log &

**22. Daemon, Service**

1) Daemon : 항상 실행이 되고 있다.

$sudo apt-get install apache2

$cd /etc/init.d/ : demon들이 위치하는 directory임

2) Demon용 서비스를 키고,끌 경우

$sudo service apache2 start :

$ps aux | grep apache2

$sudo service apache2 stop : 끄는 명령어

3) $cd \etc ; $rc.tabkey

cd rc3.d/ : S02apache2-> ../init.d/apache2

$./S02apache2

**23. cron : Time based job schedule**

1) 정기적으로 어떠한 작업을 실행하여야 하는 경우

$crontab -e : google에서 crotab exp로 검색

1/\* \* \* \* \* date >> date.log

$date >> date.log

$cat date.log

$crontab -l

$cd ~ ; ls

$tail -f date.log

$cron -e

\*/1 \* \* \* \* date 1>> date.log 2>&1: 표준 error를 표준 출력으로

2) 10000명에게 -mail을 보내는 경우 사용자가 전송하고 서버에서는 저장 후

전송자에게 response를 주고 서버 side에서 background로 전송을 하는 경우

cron을 사용함

**24 shell을 시작할 때 특정한 명령어를 실행하는 방법**

1) alias - https://www.cyberciti.biz/tips/bash-aliases-mac-centos-linux-unix.html

alias l ='ls -al'

alias .. = 'cd..'

2) shell을 열었을때 이러한 기능을 사용하려면

$echo $SHELL : /bin/bash

$cd ~

$nano .bashrc

echo 'Hi bash'

$

**25. 다중 사용자**

1) id : 나는 누구인가?

$id

2) who : 누가 접속해 있는가?

**26 관리자와 일반 사용자**

1) Root user (Super user)

일시적으로 권한을 위임 : $sudo apt-get update;

~$ : 일반유저, #: superuser

$su - root : 변경

Password:

root@ubunu:~# id : super user로 변경된 것임

root@ubunu:~# exit

$sudo passwd -u root : root password를 unlock

$su - root : super user 권한

$sudo passwd -l root : root password을 lock

$su - root

2) 일반사용자의 추가 : unix add user google검색

$sudo useradd -m duru

$su - duru

$sudo passwd duru

<sudo 권한을 줄려면>

$sudo adduser hduser sudo 또는

$sudo usermod -a -G sudo duru

**Steps to create a new sudo user on Ubuntu**

1. First add the user, run: sudo adduser <UserNameHere>
2. Add the user to sudo group by typing the command in terminal for Ubuntu version 12.04 and above: sudo adduser <UserNameHere> sudo
3. In older version of Ubuntu (version 12.04 and older), run: sudo adduser <UserNameHere> admin
4. Verify it: id <UserNameHere>

https://www.cyberciti.biz/faq/how-to-add-delete-grant-sudo-privileges-to-users-on-freebsd-unix-server/

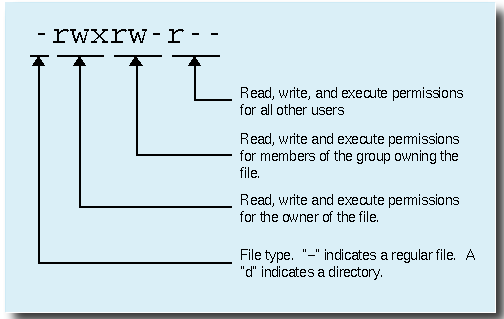
**Linux: Delete / Remove User Account**

userdel userName  
userdel [options] userName  
userdel -r userName

**27 권한(Permission)**

\* https://help.ubuntu.com/community/FilePermissions

\* https://en.wikipedia.org/wiki/Chmod

****

Here are a few examples of chmod usage with letters (try these out on your system).

First create some empty files:

user@host:/home/user$ touch file1 file2 file3 file4

user@host:/home/user$ ls -l

total 0

-rw-r--r-- 1 user user 0 Nov 19 20:13 file1

-rw-r--r-- 1 user user 0 Nov 19 20:13 file2

-rw-r--r-- 1 user user 0 Nov 19 20:13 file3

-rw-r--r-- 1 user user 0 Nov 19 20:13 file4

Add owner execute bit:

user@host:/home/user$ chmod u+x file1

user@host:/home/user$ ls -l file1

-rwxr--r-- 1 user user 0 Nov 19 20:13 file1

Add other write & execute bit:

user@host:/home/user$ chmod o+wx file2

user@host:/home/user$ ls -l file2

-rw-r--rwx 1 user user 0 Nov 19 20:13 file2

Remove group read bit:

user@host:/home/user$ chmod g-r file3

user@host:/home/user$ ls -l file3

-rw----r-- 1 user user 0 Nov 19 20:13 file3

Add read, write and execute to everyone:

user@host:/home/user$ chmod ugo+rwx file4

user@host:/home/user$ ls -l file4

-rwxrwxrwx 1 user user 0 Nov 19 20:13 file4

user@host:/home/user$

chmod with Numbers

Usage: chmod {options} filename

|  |  |
| --- | --- |
| **Options** | **Definition** |
| #-- | owner |
| -#- | group |
| --# | other |
| 1 | execute |
| 2 | write |
| 4 | read |

**Numerical permissions**

|  |  |  |
| --- | --- | --- |
| **#** | **Permission** | **rwx** |
| 7 | read, write and execute | rwx |
| 6 | read and write | rw- |
| 5 | read and execute | r-x |
| 4 | read only | r-- |
| 3 | write and execute | -wx |
| 2 | write only | -w- |
| 1 | execute only | --x |
| 0 | none | --- |

Here are a few examples of chmod usage with numbers (try these out on your system).

First create some empty files:

user@host:/home/user$ touch file1 file2 file3 file4

user@host:/home/user$ ls -l

total 0

-rw-r--r-- 1 user user 0 Nov 19 20:13 file1

-rw-r--r-- 1 user user 0 Nov 19 20:13 file2

-rw-r--r-- 1 user user 0 Nov 19 20:13 file3

-rw-r--r-- 1 user user 0 Nov 19 20:13 file4

Add owner execute bit:

user@host:/home/user$ chmod 744 file1

user@host:/home/user$ ls -l file1

-rwxr--r-- 1 user user 0 Nov 19 20:13 file1

Add other write & execute bit:

user@host:/home/user$ chmod 647 file2

user@host:/home/user$ ls -l file2

-rw-r--rwx 1 user user 0 Nov 19 20:13 file2

Remove group read bit:

user@host:/home/user$ chmod 604 file3

user@host:/home/user$ ls -l file3

-rw----r-- 1 user user 0 Nov 19 20:13 file3

Add read, write and execute to everyone:

user@host:/home/user$ chmod 777 file4

user@host:/home/user$ ls -l file4

-rwxrwxrwx 1 user user 0 Nov 19 20:13 file4

user@host:/home/user$

**chmod with sudo**

Changing permissions on files that you do not have ownership of: (**Note** that changing permissions the wrong way on the wrong files can quickly mess up your system a great deal! Please be careful when using **sudo**!)

user@host:/home/user$ ls -l /usr/local/bin/somefile

-rw-r--r-- 1 root root 550 2005-11-13 19:45 /usr/local/bin/somefile

user@host:/home/user$

user@host:/home/user$ sudo chmod o+x /usr/local/bin/somefile

user@host:/home/user$ ls -l /usr/local/bin/somefile

-rw-r--r-x 1 root root 550 2005-11-13 19:45 /usr/local/bin/somefile

user@host:/home/user$

**28. 그룹**

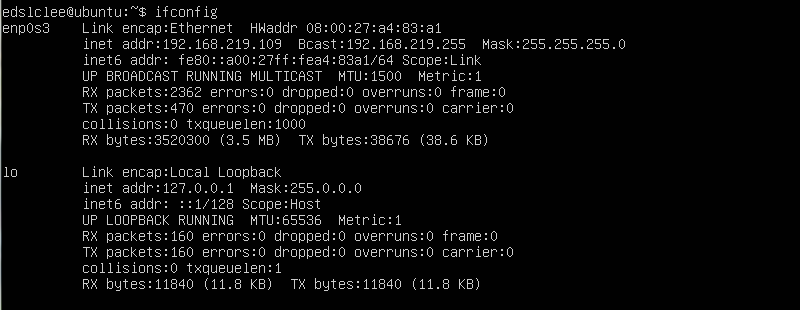
**29. 인터넷, 네트워크,서버**

1) 자신의 IP Address를 알 수 있는 방법

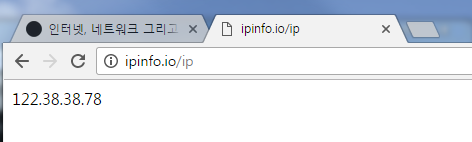
**- $ip addr**



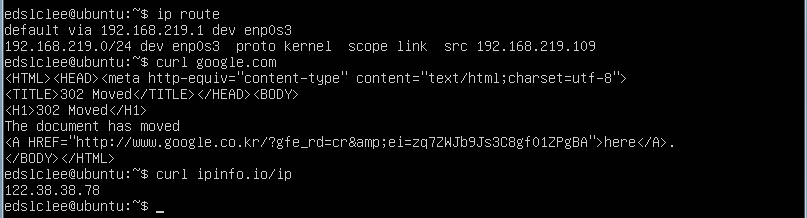
**- $ifconfig**



**- ipinfo.io/ip**

****

**linux : $curl google.com; curl ipinfo.io/ip**

****

**- default IP address : $ip route**

****

2) 외부 IP(ipinfo.io/ip)는 122.38.38.78이고 내부 IP의 default gateway(ip route)는 192.168.219.1 서버가 설치된 내부 IP(ifconfig)는 192.168.219.109임

****

122.38.38.78:9000

192.168.219.1

9000-> 192.168.219.109:80

Private Address

Public Address

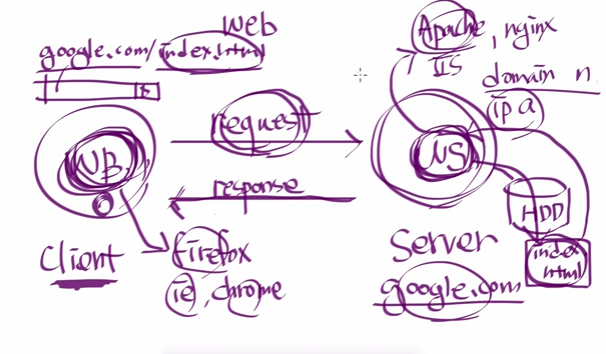
192.168.219.104

192.168.219.106

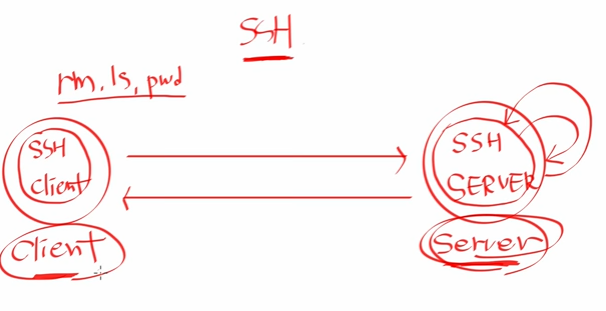
122.38.38.78

192.168.219.109

**30. 웹서버(Apache)**



**31. 원격제어(ssh)**

****

1) Sever설치

- $sudo apt-get install openssh-server

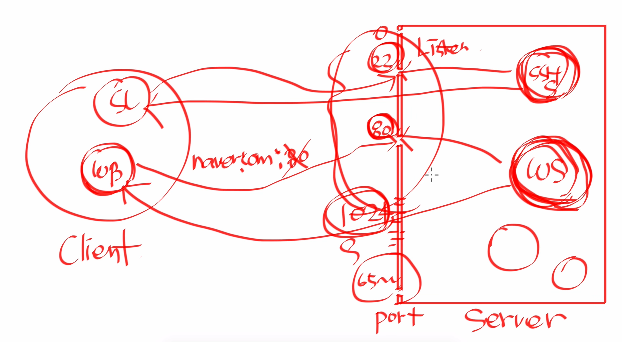
- $sudo service ssh start

- $sudo aux | grep ssh

2) client

windows Putty설치

**32. port**

****

1) Port개요

$sudo nano /etc/ssh/sshd\_config

port 22 -> 2222

$sudo service ssh restart

client에서 2222로 접속해야 함.

2) port forwading



**33. Domain**

**34. 서버간 동기화(rsync)**

$mkdir rsync

$cd rsync

$mkdir source dest

$cd source

$touch test{1..10} : 한꺼번에 10개의 파일을 생성함 test1 ~ test10까지

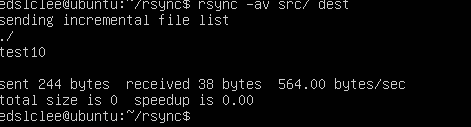
$cd ..

$rsync source\ dest : source밑의 도든 파일을 des로 Rsync한다

이 상태에서 dest의 test10 파일을 삭제하고

$rsync -av source\ dest를 하면

test10만 복사한다 : 즉 증분된(incremental) 부분만 복사함



원본과 복사본이 같으면 아무것도 안한다.

rsync -a : archive mode로 동작한다.

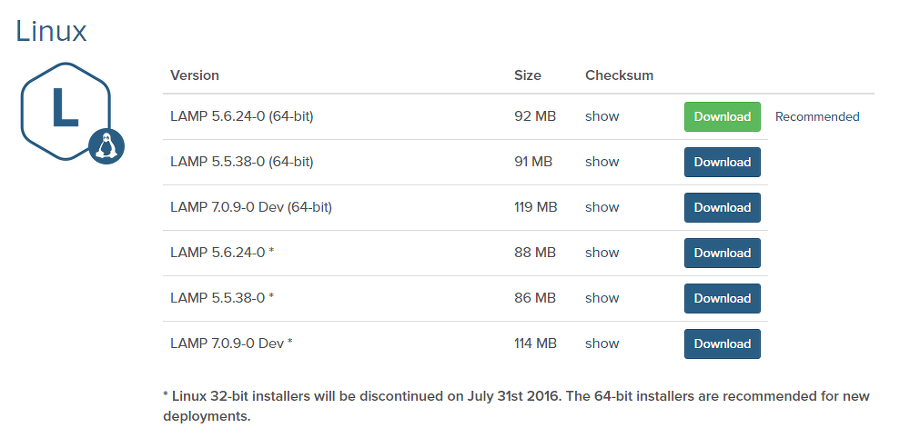
- remote

$rsync -aZP ~/rsync/src/ k8805@192.168.0.65:~/rsync/dest/

password:

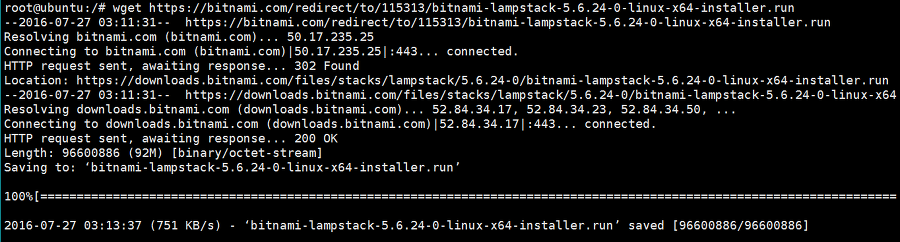
**35. Bitmnami설치**

https://bitnami.com/stack/lamp/installer



다운 받을 파일의 버튼을 오른쪽 클릭한 다음 링크 주소 복사를 한다. 터미널 창으로 들어가 wget 명령어를 사용하여 다운받는다.wget 복사한 링크 주소 붙여넣기

wget https://bitnami.com/redirect/to/115313/bitnami-lampstack-5.6.24-0-linux-x64-installer.run



chmod 명령어를 사용하여 권한을 변경해 준다.

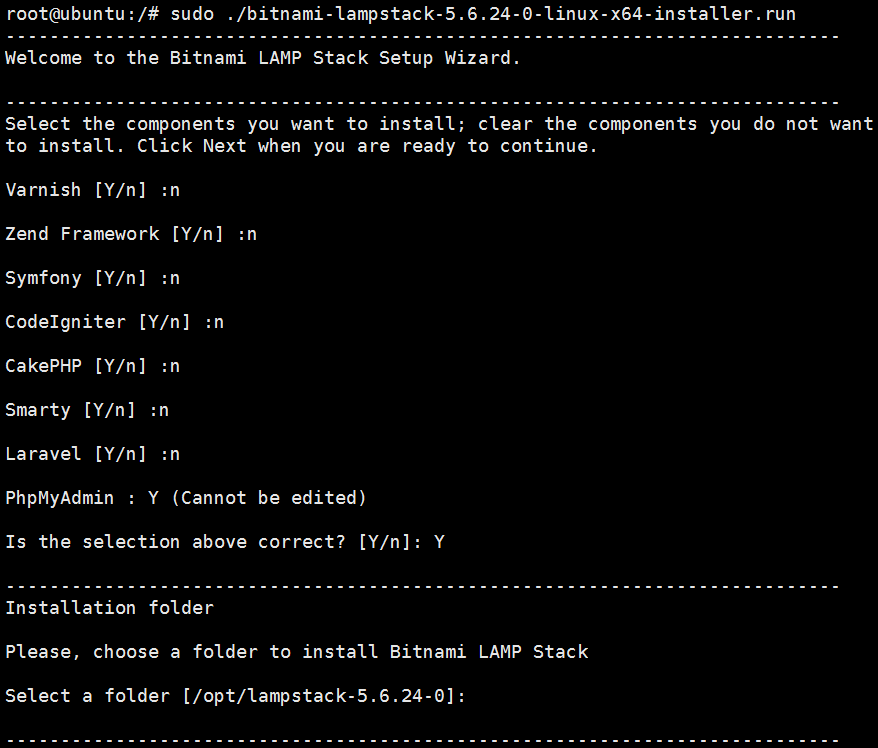
sudo chmod 755 bitnami-lampstack-5.6.24-0-linux-x64-installer.run

lamp를 설치 한다.

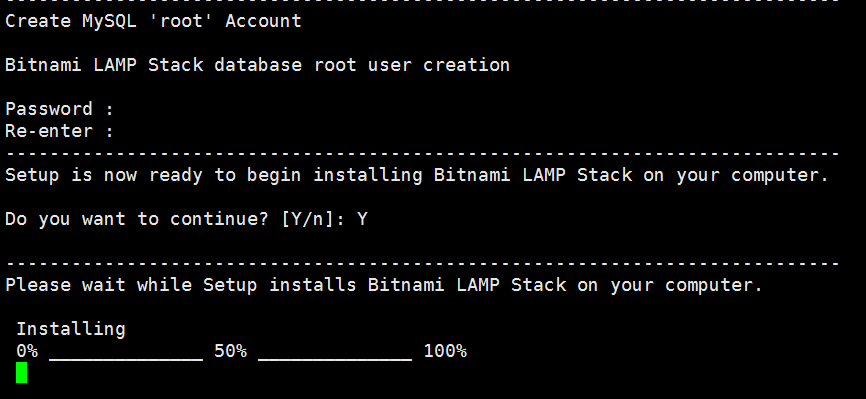
sudo ./bitnami-lampstack-5.6.24-0-linux-x64-installer.run

설치할 때 PhpMyAdmin을 제외한 나머지는 설치 하지 않는다. 전부 n 를 해준다.

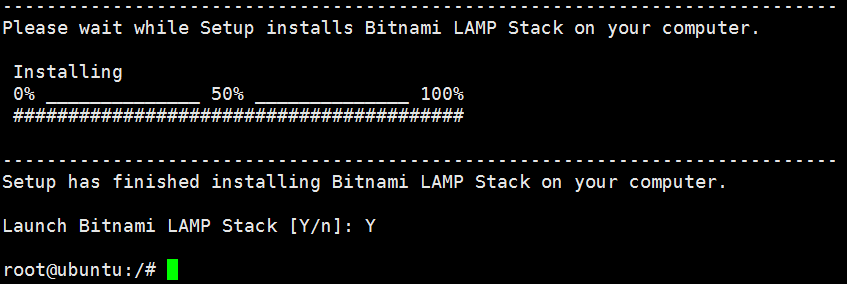
Bitnami의 설치 위치를 변경하지 않는다면 엔터 해준다. ( 기본은 /opt/lampstack-버전 )



MySQL의 root 사용자의 비밀번호를 설정해 준다.



설치 완료하면 실행 시켜 준다~



apache 설정 파일 : Bitnami 설치경로/apache2/conf/httpd.conf

document root : Bitnami 설치경로/apache2/htdocs/

**36. server의 IP가 변동되는 경우**

1) IPv6로 변경되지 않도록 하는 방법

cd /etc/default

명령으로 해당 디렉토리로 이동. sudo vi grub  이나 sudo gedit grub , sudo nano grub  같은 명령으로 해당 파일을 수정. 수정해야 할 곳은

GRUB\_CMDLINE\_LINUX\_DEFAULT="quiet splash ipv6.disable=1"

다음, grub.cfg 를 생성해 준다.

sudo grub-mkconfig -o /boot/grub/grub.cfg

명령으로 해당 파일을 생성. 제대로 설정 되었는지 확인하려면, 재부팅시에 grub 이 실행되었을때 'e' 키를 눌러서 우리가 추가한 옵션이 잘 추가되어 있는지 확인 해 볼 수 있다.또한 ipv6 가 제대로 비활성화 되어있는지 확인해 보려면 부팅 후 에 '/proc/sys/net'  디렉토리를 확인해 보면 된다.원래는 여기에 'ipv6' 디렉토리가 있으나 위처럼 설정하고 확인해 보면 아예 ipv6 디렉토리가 존재하지 않는다.

2) 서버의 Ethernet IP가 변경될 경우 변경이 필요한 것들

2-1) Web Server의 주소: ifconfig, ip addr의 주소로

2-2) PhpMyadmin :

- putty의 접속변경 : 변경된 서버 IP로 변경

- http://127.0.0.1:8888/phpmyadmin/

**37. 서버제어**

1) Apache, MySQl 기동

sudo /opt/lampstack-5.6.30-1/ctlscript.sh start

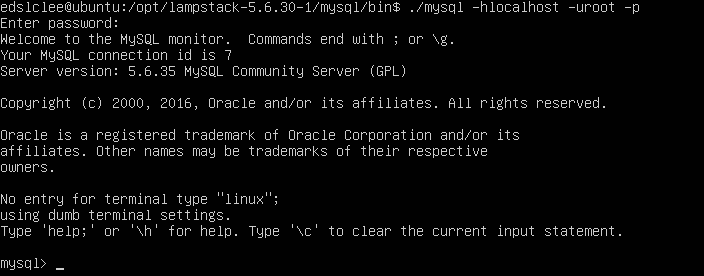
sudo /opt/lampstack-5.6.30-1/ctlscript.sh restart mysql

sudo /opt/lampstack-5.6.30-1/ctlscript.sh restart apache

Obtain current status of all services:

sudo /opt/lampstack-5.6.30-1/ctlscript.sh status

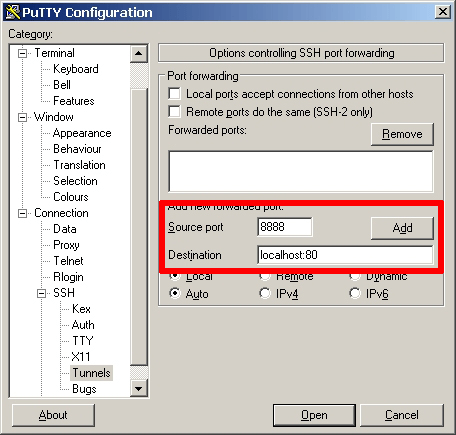
MYSQL



2) putty로 ssh session을 열고 phpmyadmin 접속

To access the application using your Web browser, create an SSH tunnel, as described below.

* Download PuTTY and make sure you can log in to the virtual machine console with it following the instructions in [the FAQ](https://docs.bitnami.com/virtual-machine/faq#how-to-connect-to-the-server-through-ssh). Once you have confirmed you are able to log in successfully, log back out.
* Reconnect to the virtual machine using PuTTY, this time adapting the steps to include an additional SSH tunnel. When configuring the new SSH session in PuTTY, additionally navigate to the "Connection -> SSH -> Tunnels" section and create a secure tunnel by forwarding port 80 on the virtual machine to port 8888 on the local host (127.0.0.1 or localhost).
* Click the "Add" button to add the secure tunnel configuration to the session. Here is an example:

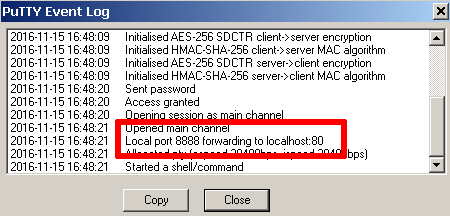
[](https://docs.bitnami.com/images/img/infrastructure/common/create-ssh-tunnel-8888-1822d77b.png)

* Go back to the "Session" section and save your changes by clicking the "Save" button.
* Click the "Open" button to open an SSH session to the virtual machine. The SSH session will now include a secure SSH tunnel between the two specified ports.

While the tunnel is active, you should be able to access the phpMyAdmin console through the secure SSH tunnel you created, by browsing to http://127.0.0.1:8888/phpmyadmin.

To log in, use username root for MySQL. The default password is bitnami.

If you are unable to access phpMyAdmin, verify that the SSH tunnel was created by checking the PuTTY event log (accessible via the "Event Log" menu):

[](https://docs.bitnami.com/images/img/infrastructure/common/create-ssh-tunnel-log-409dee64.png)

3) web Browser에서 127.0.0.1:8888/phpmyadmin을 입력 연결됨.

**38. SSH Key를 이용하여 Git로 전송하기**

Git bash에서 ssh-keygen :

c:/user/.ssh/rsa-pub에서 Rsa-pub를 copy하여 github에 setup에서 등록