

강의 소개

1. 강의 목표

- 1) Linux 기반 개발 및 분석 Framework(Hadoop, Spark 등)의 원활한 사용을 위해,
- 2) Linux OS 에 대한 기본 배경 지식을 인지하고,
- 3) 실습을 통해 Linux 기본 명령어 및 Shell Script 를 다뤄봄으로써,
- 4) Linux 환경에서의 다양한 기법을 활용할 수 있는 기반 역량을 확보하는 것이 강의의 목표임

2. 강의 Agenda

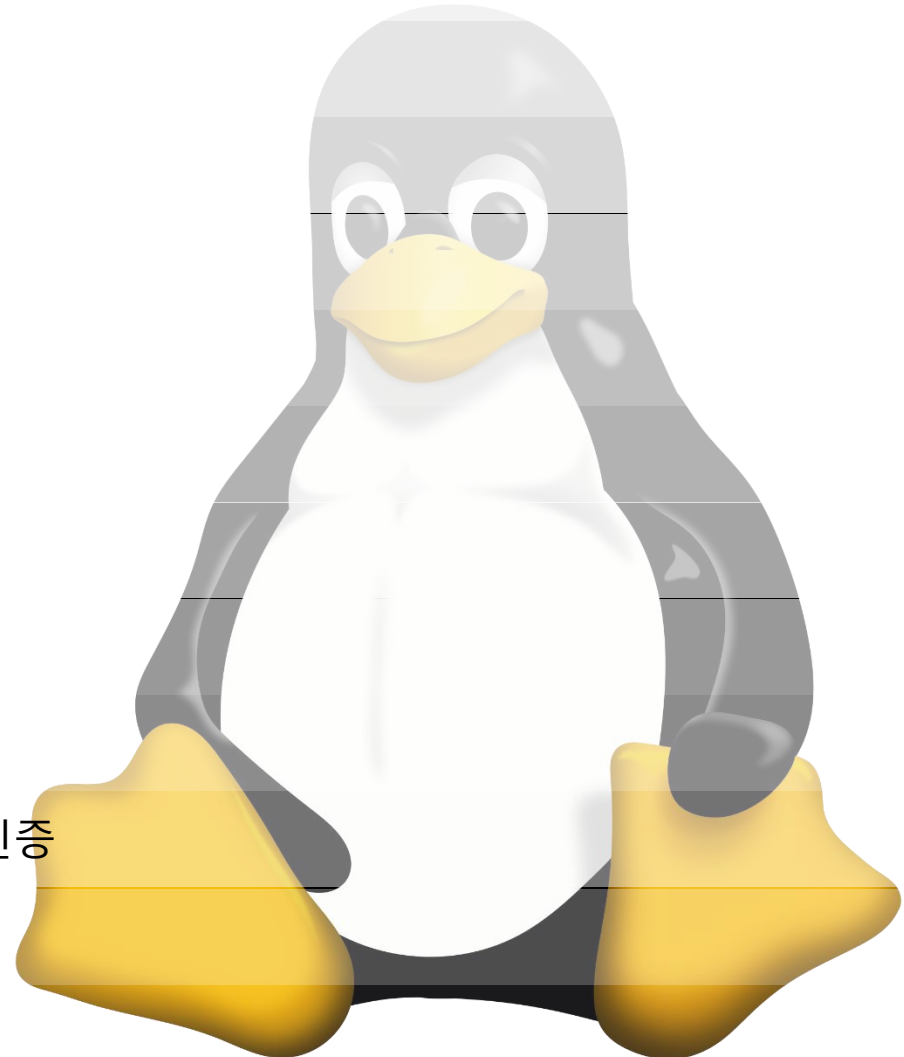
- 1) Linux에 대한 기본적인 발전 과정과 사용현황을 살펴보고,
- 2) 각 사용 목적 별 명령어들을 소개하고 관련 사용예를 살펴보고,
- 3) 학습한 명령어를 조합하여 응용할 수 있는 시간을 가지고,
- 4) 전체적인 Linux의 구조와 사용법을 자연스럽게 익힐 수 있도록 구성

3. 강의 진행

-Seoul Coding

Table of Contents

1. Linux 소개
2. Linux 실습 환경 및 유용한 팁
3. File 및 Directory 관리 명령어
4. User 및 Permission 관리 명령어
5. Network 관리 명령어
6. Device 관리 명령어
7. Environment Variable 설정
8. APT Package 관리 도구
9. Process 및 Resource 관리 명령어
10. Remote Access 및 SSH 키 생성을 통한 인증
11. Log 관리 파일 및 명령어
12. AWS(Amazon Web Service) 이용 방법



1. Linux 소개

- 1) OS와 Linux
- 2) Windows의 역사
- 3) Unix/Linux의 역사
- 4) GNU 프로젝트 및 GPL
- 5) Linux의 역사 및 커널
- 6) Linux의 특징
- 7) Linux 배포판

OS와 Linux

- **운영체제(operating system, OS)**

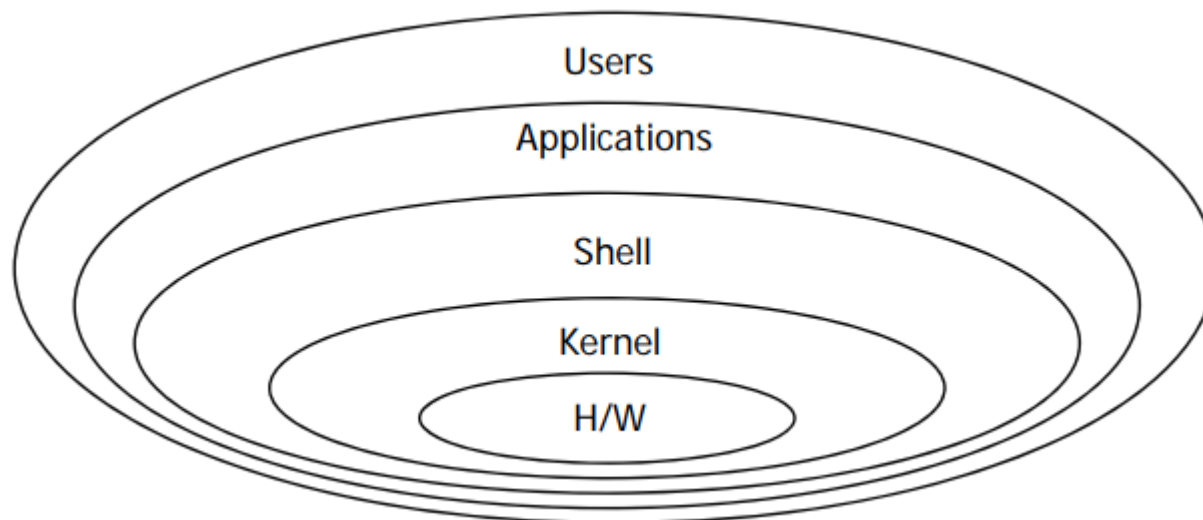
- 컴퓨터 자원을 효율적으로 관리하며 사용자와 컴퓨터 사이에 인터페이스를 제공

- **kernel**

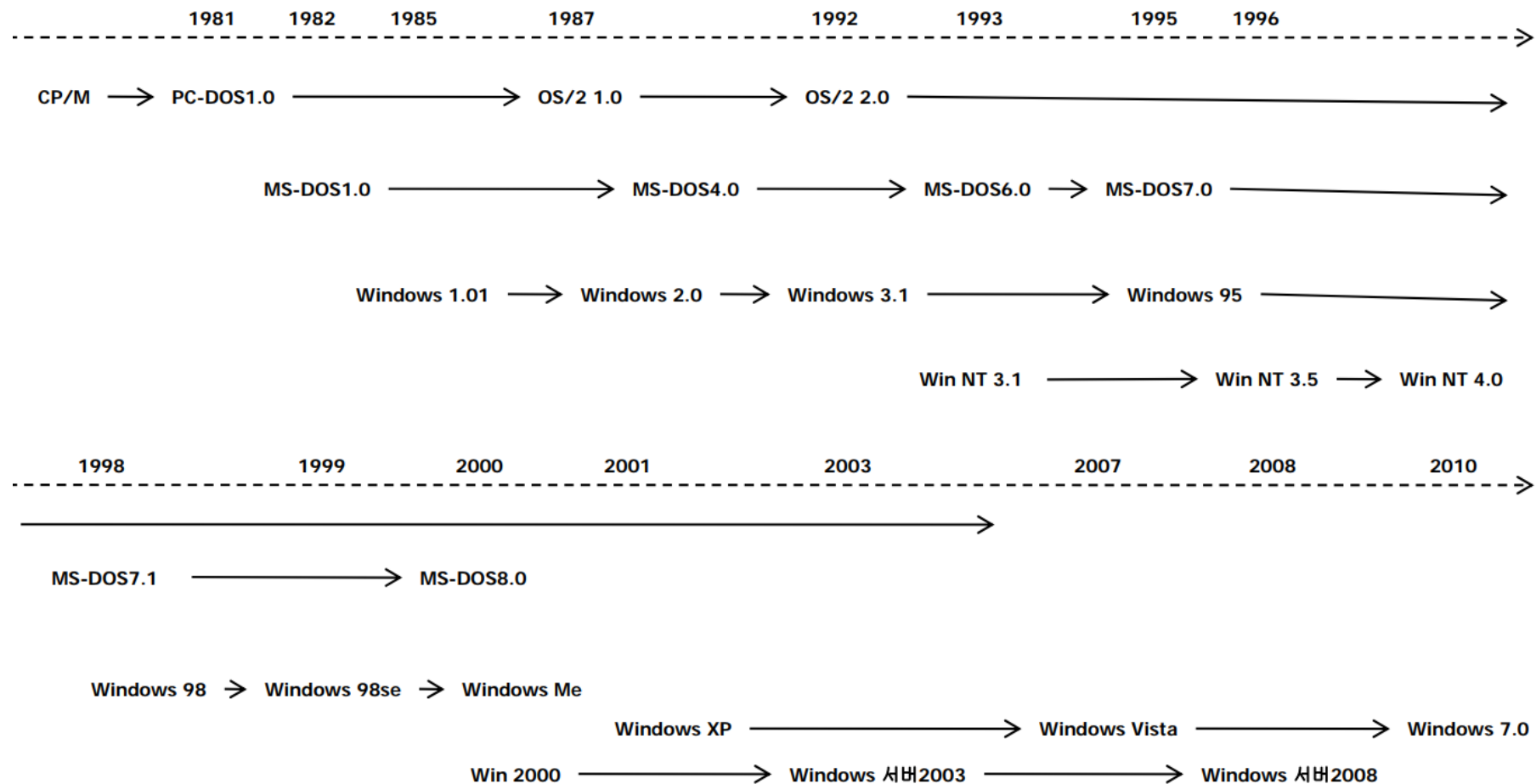
- 운영체제의 핵심부분(프로세스관리, 메모리관리, I/O 시스템 관리, 파일시스템관리 등)

- **shell**

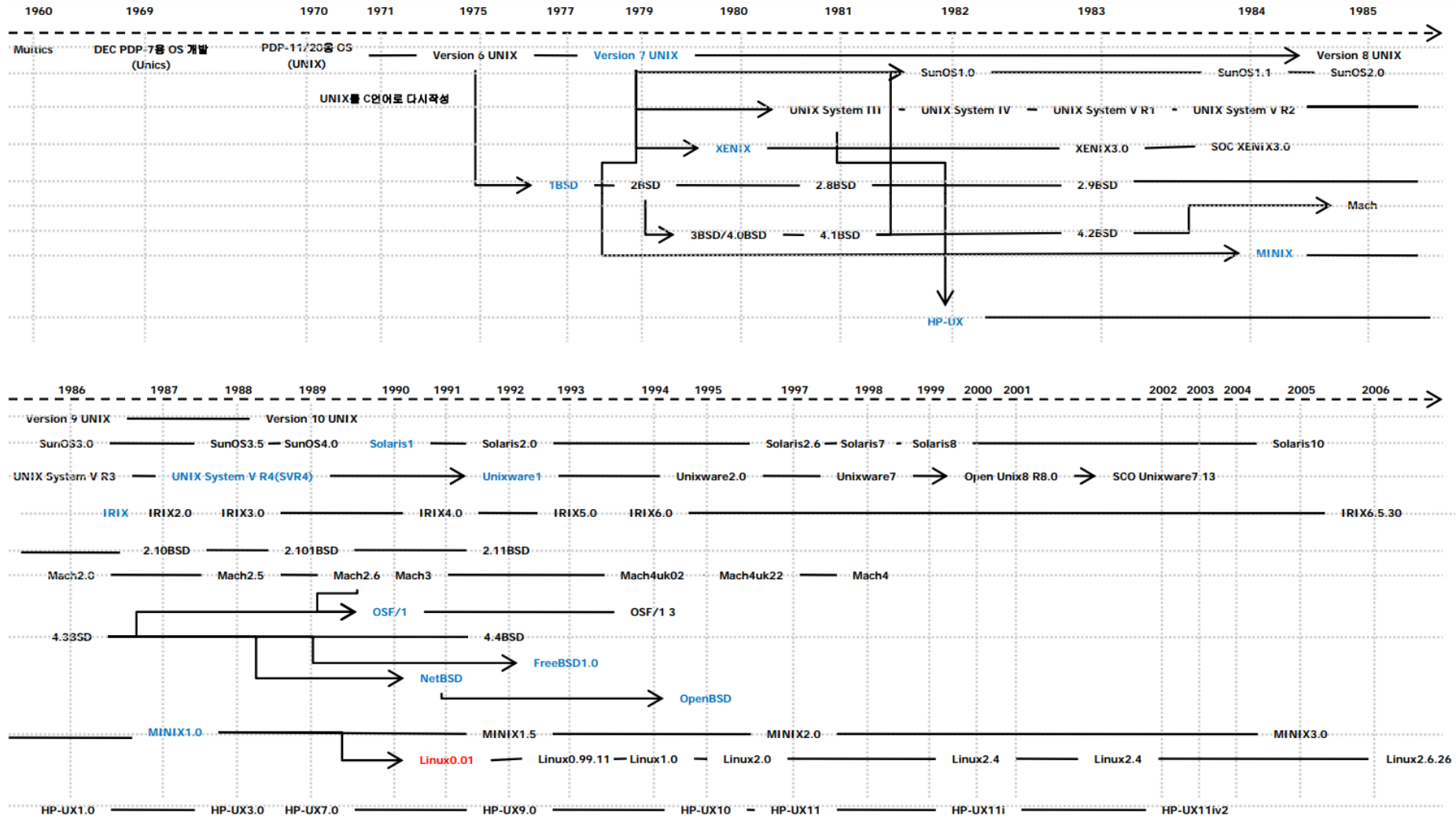
- 사용자와 운영체제의 내부 kernel 사이의 인터페이스 역할 (명령어 해석기)



Windows의 역사



Unix/Linux의 역사



GNU 프로젝트 및 GPL

■ GNU (GNU is Not Unix)

- <http://www.gnu.org/>
- 80년대 리처드 스톨만(Richard Stallman)에 의하여 시작
- GNU 프로젝트의 목적 : 자유 소프트웨어로만 구성된 하나의 완전한 Unix 시스템을 만드는 것
- 리눅스는 GPL라이선스에 따라 무료로 배포 및 수정되어 사용할 수 있는 운영체제
- Linux에 많은 유용한 유틸리티를 포함시키게 됨

■ GNU GPL (GNU General Public License)

- 누구나 자유롭게 사용, 변경, 배포가 가능
- 다양한 리눅스 배포판 존재
- 변경 사항을 포함해서 재판매하는 것은 허용하나 소스는 공개해야 함.
- 프로그래머는 자신의 소프트웨어로 발생하는 어떤 위험이나 손해에 대한 법률적 책임이 없음.
- (참고)
 - 배포판과 함께 제공되는 S/W : 유료 및 무료 가능
 - 현재의 기업
 - 프로그램 자체에 대해서 돈을 받기보다,
 - 서비스의 질과 시간적 양에 따라 차별적으로 돈을 받는 것을 추구
 - 소프트웨어에 대한 공식적인 지원, 문제발생 시 책임 문제 고려하여,
 - 상황에 따라 상용 리눅스 사용

Linux의 역사 및 커널



- 리처드 스톨만(Richard M. Stallman)

- MIT인공지능 연구소의 연구원
- 기업을 중심으로 소스를 공개하지 못 하도록 하는 분위기, 기술을 사업화 하려는 조류에 대한 반감

- 1983년

- GNU(GNU is Not Unix) 프로젝트를 시작

- 1985년

- Emacs에 대한 사용자들에 대한 관심이 높아짐.
- 자유 소프트웨어 재단(FSF : Free Software Foundation)을 설립 (GNU프로젝트 운영을 위해)

- 1990년

- GNU프로젝트는 시스템 라이브러리, 컴파일러, 텍스트 에디터, 셸 등의 시스템의 핵심적인 부분을 거의 완성시켰다. (단, 운영체제에서 핵심이 되는 커널이 빠져있는 상태)
- Mach를 기반으로 한 커널 허드(Hurd)를 개발하기 시작
- 마하가 복잡한 구조를 가지고 있어 Hurd의 개발이 지체됨

Linux의 역사 및 커널



■ 리누스 토발즈(Linus Benedict Torvalds)

- 앤드류 타넨바움 (Andrew S. Tanenbaum) 교수가 운영체제 디자인을 가르치기 위해 만든 교육용 유닉스인 미닉스(MINIX)에서 아이디어를 얻어, 독자적으로 리눅스를 개발 (타넨바움은 미닉스를 다른 사람이 함부로 수정하지 못하도록 제한)
 - 뉴스그룹(comp.so.minix)을 통해 리눅스를 개발하고 있음을 알림
-
- 1991년 9월
 - 리눅스의 등장 : 0.01버전 개발 (미공개)
 - 1991년 10월
 - 0.02버전 : 뉴스그룹에 첫 공식적인 발표
 - 10명이 다운로드, 그 중 5명이 버그를 수정 및 개선하여 보내옴
 - 1992년 3월
 - 0.95 버전 (GUI와 Intel x86 지원)
 - 리처드 스톨만과 FSF는 리눅스를 GNU 커널로 채택
 - 리눅스는 GNU C 컴파일러(gcc)로 컴파일한 응용프로그램 증가
 - 1994년
 - 1.0 버전 (네트워크 기능 추가), 운영체제실습2(Linux)

Linux의 역사 및 커널

- 1996년 6월
 - 2.0 버전 (SMP 기능 추가)
- 1999년 1월
 - 2.2 버전 (최대 16개의 CPU 지원과 최대 동시접속 사용자 2,048명까지 지원)
- 2001년 1월
 - 2.4 버전
- 2018년 현재 안정버전 : 4.15

(참고)

▪ 리눅스 이름의 유래 :

- 아리람케라(ftp.funet.fi사이트의 운영자)는 사람이 LINUs' miniX => LINUX폴더를 만들어 토발즈에게 제공한 것이 유래 (원래 : 토발즈는 Freax(프릭스)로 하렘했다)

▪ 리눅스 로고

- 1996년 래리윙이 창조한 리눅스의 마스코드(Tux(턱스) : Torvalds UniX => TUX)
- 리누스는 펭귄의 모습이 청어를 배불리 먹고 포만감에 젖어 편안히 앉아 있는 자세의 정다운 모습이라고 묘사

Linux의 특징

▪ Linux의 특징

- 다중 플랫폼(Multi-platform)지원
 - Intel CPU (i386), Digital Alpha, Sun Sparc, Sparc64, PowerPC 등
- 다양한 하드웨어 장치 지원
- 네트워크 기능 제공
- 이식성이 뛰어나
 - C언어 기반으로, 프로그래밍과 porting이 용이
- 유닉스의 특징
 - 다중 사용자(Multi-user)가 동시에 사용할 수 있는 환경을 제공
 - 다중 작업(Multi-Tasking) 환경 제공
 - 트리 형태의 계층적 구조로 된 파일시스템
 - 풍부한 소프트웨어 개발환경 제공
 - 거의 모든 프로그래밍 언어 제공
 - 강력한 네트워킹 기능 제공

Linux의 특징

Windows vs Linux



Open Source

Free

Free Software

Live CD Distribution

Secure

NO

Low Hardware Cost

Customizable add features

Closed Source

Cost 150\$~450\$

Cost software

NO

Insecure

Virus, Malware

High Hardware Cost

Not Customizable

Linux 배포판 (1/2)

Linux World Map 2.0



< Source : <http://www.dedoimedo.com> >

Linux 배포판 (2/2)

- 약 300여종의 배포판이 존재함
- 배포판 선택시 고려사항

- 사용자의 목적/용도, 사용 경험, 데스크탑/서버 등을 고려하여 선택이 달라질 수 있다.

주요 리눅스 배포판

배포판	설명
슬랙웨어 (Slackware)	최초의 리눅스 배포판 가운데 하나로, 리눅스 광들에게 인기가 높음
레드햇(Red Hat)	상업 비즈니스용 배포판으로 인터넷 서버에 주로 사용됨
페도라 (Fedora)	레드햇에서 파생되었으며 일반 사용자용으로 설계됨
젠투 (Gentoo)	고급 리눅스 사용자를 위한 배포판으로 리눅스 소스코드만 포함하고 있음
오픈수세 (openSUSE)	비즈니스 및 일반 사용자용으로 다양한 배포판이 있음
데비안 (Devian)	리눅스 전문가들에게 인기가 높으며 상용 리눅스 제품임

전문화된 리눅스 배포판

배포판	설명
센트OS (CentOS)	레드햇 엔터프라이즈 리눅스 소스코드로 만든 무료 배포판
우분투 (Ubuntu)	학교와 가정 사용자들을 위한 배포판
PC리눅스 (PCLinuxOS)	가정 및 사무실 사용을 위한 무료 배포판
민트 (Mint)	홈 엔터테인먼트 사용자를 위한 무료 배포판
다인:볼릭 (dyne:bolic)	오디오와 MIDI 애플리케이션을 위해 설계된 무료 배포판
퍼피 리눅스 (Puppy Linux)	구형 PC에서도 잘 실행되는 가벼운 무료 배포판

2. Linux 실습환경 및 유용한 팁

- 1) Linux 실습환경 구성 방법
- 2) Virtualbox에 Linux 설치
- 3) Virtualbox Network 연결 방식
- 4) Linux 부팅 및 종료
- 5) 자주 사용하는 유용한 팁

Linux 실습환경 구성 방법

▪ Local System에 설치하여 사용

- HDD 파티션을 분리하여 windows 와 Linux 두개의 시스템을 동시에 사용 가능
- 멀티 OS 부팅 설정 방법과 HDD 파티셔닝 방법 습득 필요
- 실제 서버를 구성할 때는 Linux 하나만 설치하여 사용

▪ Virtual Machine Manager를 활용하여 사용 => 본 강의 실습 방법

- VMware, VirtualBox 등의 VMM을 이용하여 Linux 설치 후 사용
- 다양한 OS를 설치하여 테스트할 수 있는 환경에 적합
- 간편하게 실습환경을 백업하고 복원 가능

▪ Public Cloud Service 를 활용하여 사용

- AWS와 같은 Public Cloud Service를 사용하여 Linux Machine을 생성한 후 원격접속 프로그램을 통해 접속하여 활용가능
- 가장 빠른 시간안에 설치가 가능함.
- Cloud 상에 VM Image를 저장하기 때문에 On-demand 환경에서 언제 어디서든 사용 가능

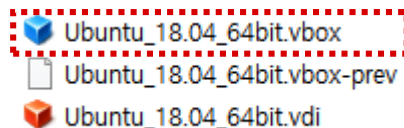
Virtualbox에 Linux 설치

■ Oracle VirtualBox 개요

- Download URL : <https://www.virtualbox.org>
- 지원되는 Platform : Windows, OS X, Linux, Solaris
- 최신 Release Version : VirtualBox 5.2.12 (2018.6.24 기준)

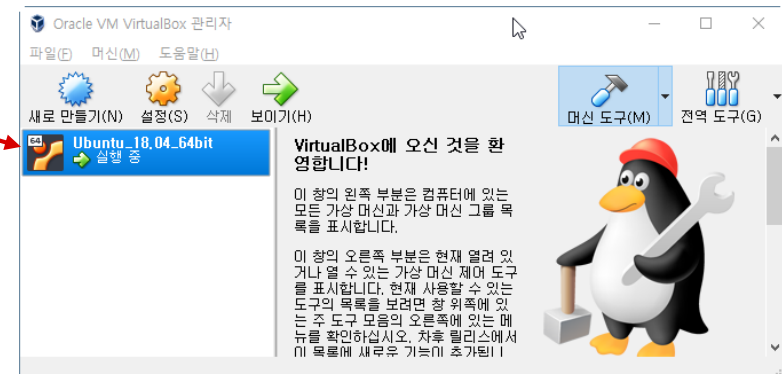
■ Ubuntu Linux 설치 (사전 배포 파일을 이용한 설치 방법)

- ① Virtualbox 최신 버전을 다운받아 설치한다. (동일 버전의 VirtualBox Extension Pack도 설치)
- ② Ubuntu 홈페이지에서 최신 버전의 Ubuntu Server ISO 파일을 다운로드 한다.
<https://www.ubuntu.com/download/server>
- ③ 사전 배포 파일의 압축을 풀고, 확장자가 .vbox 인 파일을 실행하면 VirtualBox에 Load 된다.



■ 배포된 VM Image 설정 정보

- 시스템 : 1024MB RAM, 1 CPU
- 저장소 : 40.87 GB HDD, 동적 할당
- 네트워크 : NAT (Network Address Translation)



Virtualbox Network 연결 방식

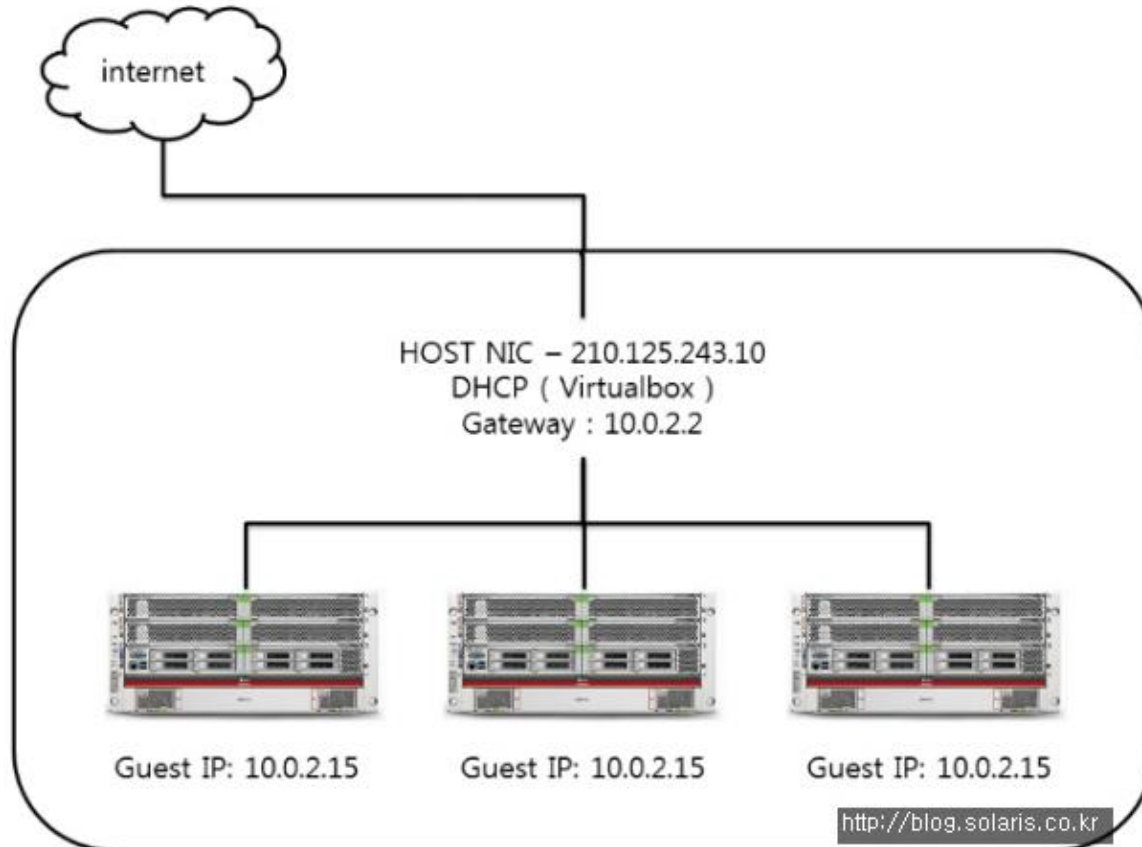
VirtualBox Network 연결 방식

- VirtualBox에서는 6가지의 네트워크 연결 방식을 지원

연결방식	설명
Not Attached	네트워크를 연결하지 않는다. NIC(Network Interface Card) 재구성 또는 네트워크 강제 연결 일시 중지를 위해 사용됨
Network Address Translation(NAT)	게스트 OS 내부에서 인터넷을 사용할 수 있는 모드로 기본설정 됨 ※ vmware의 nat와는 다르게 구성되어짐
Bridged Networking	호스트 OS의 네트워크 스택을 경유하여 NIC를 통해 연결 가능함 IP 공유기등이 존재하는 경우, IP할당에 제약이 없는 경우 등에 적합한 방식
Internal Networking	내부 가상머신들간의 통신을 지원하는 모드로 호스트나 외부 네트워크와는 단절 됨
Host-only Networking	가상머신과 호스트간의 연결을 제공하기 위해 존재하며 별도의 물리적 네트워크 카드 없이 가상네트워크 인터페이스가 호스트에 생성됨
Generic Networking	확장팩 또는 VirtualBox 내에서 제공되는 드라이버를 선택 할 수 있도록 하여 네트워크 인터페이스를 공유하는 모드로 일반적으로 사용되지 않음

Virtualbox Network 연결 방식

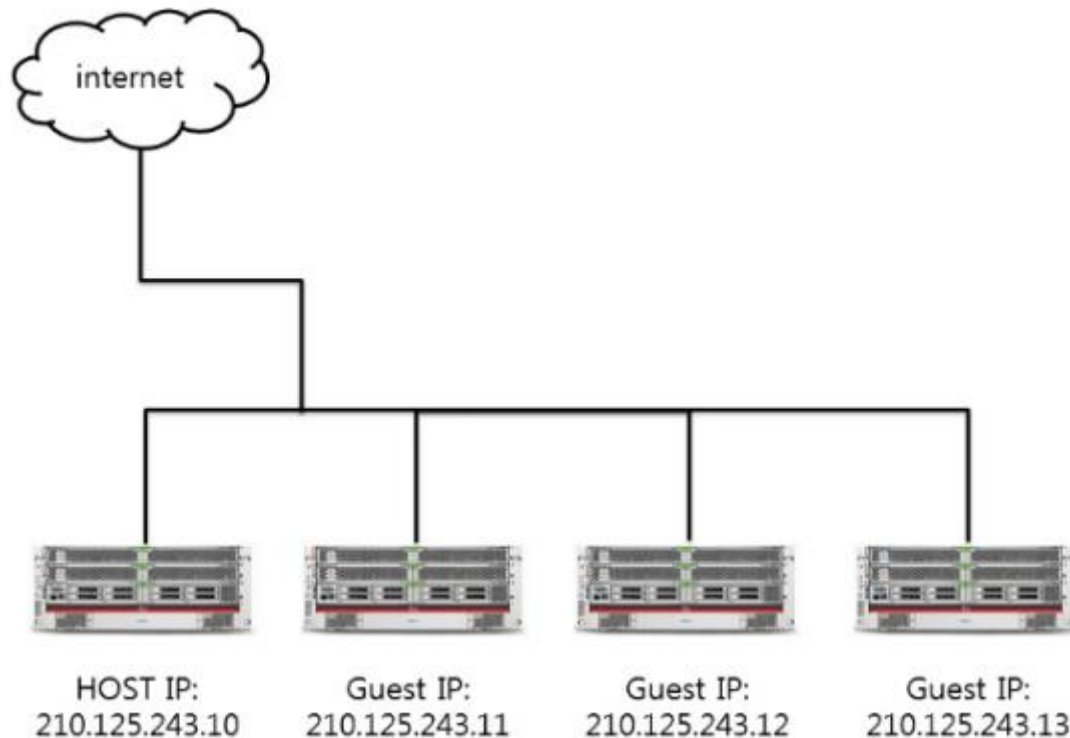
▪ NAT (Network Address Translation)



- 가상머신이 별도 가상네트워크에 할당된 10.0.2.2 게이트웨이를 경유하여 인터넷에 연결됨
- 하나의 네트워크 내에서 인터넷을 자유롭게 이용 가능하지만 각각 독립적 네트워크로 구성되어 상호간의 연결을 지원하지 않음
- DHCP를 사용하는 경우 10.0.2.15로 동일한 IP를 할당받아 사용

Virtualbox Network 연결 방식

▪ Bridged Networking

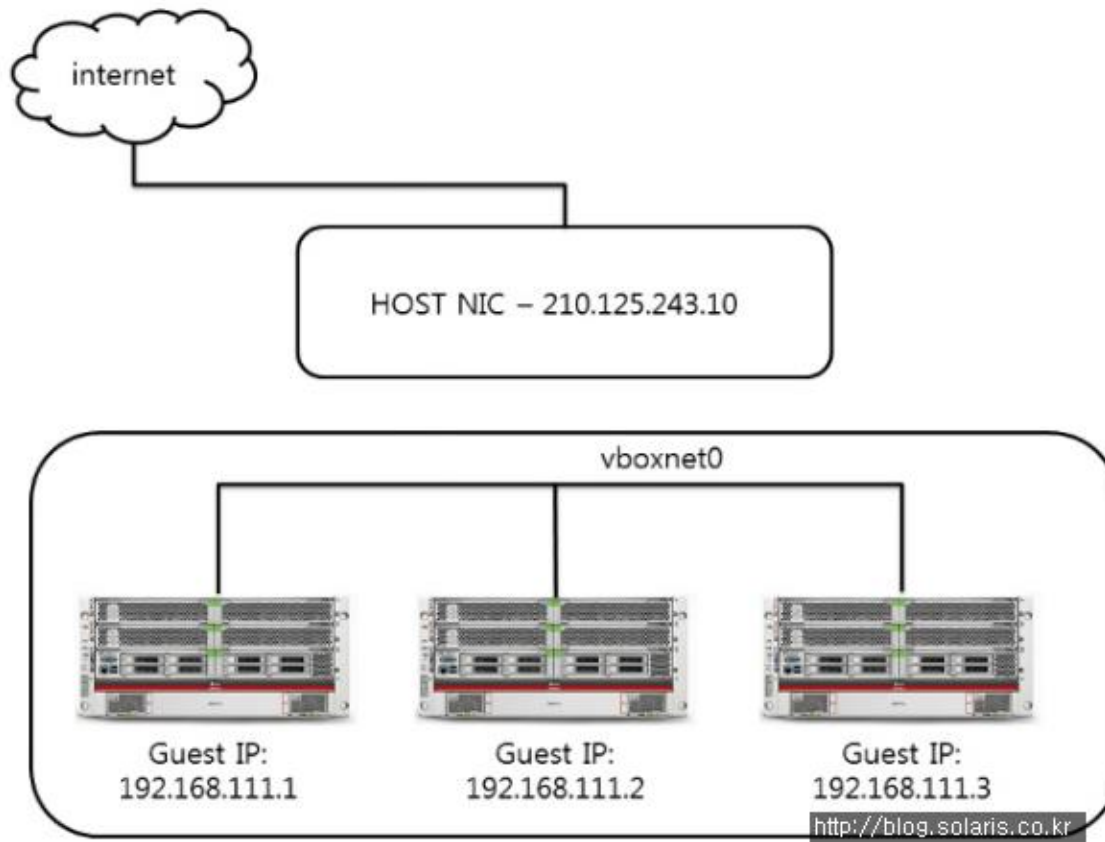


<http://blog.solaris.co.kr>

- 실제 공인 IP를 여러 개 부여하여 서비스를 제공할 수도 있고, 외부 게이트웨이를 이용하여 인터넷 이용도 가능
- IP공유기를 사용하는 경우 IP공유기 내에서도 사용 가능
- 모든 게스트와 호스트를 동일한 네트워크로 구성하며, 별도의 다른 네트워크를 설계하여 격리된 네트워크 설계도 가능
- 네트워크 임의의 구성이 불가능하거나 IP구성이 원활하지 않는 경우 사용이 제한됨

Virtualbox Network 연결 방식

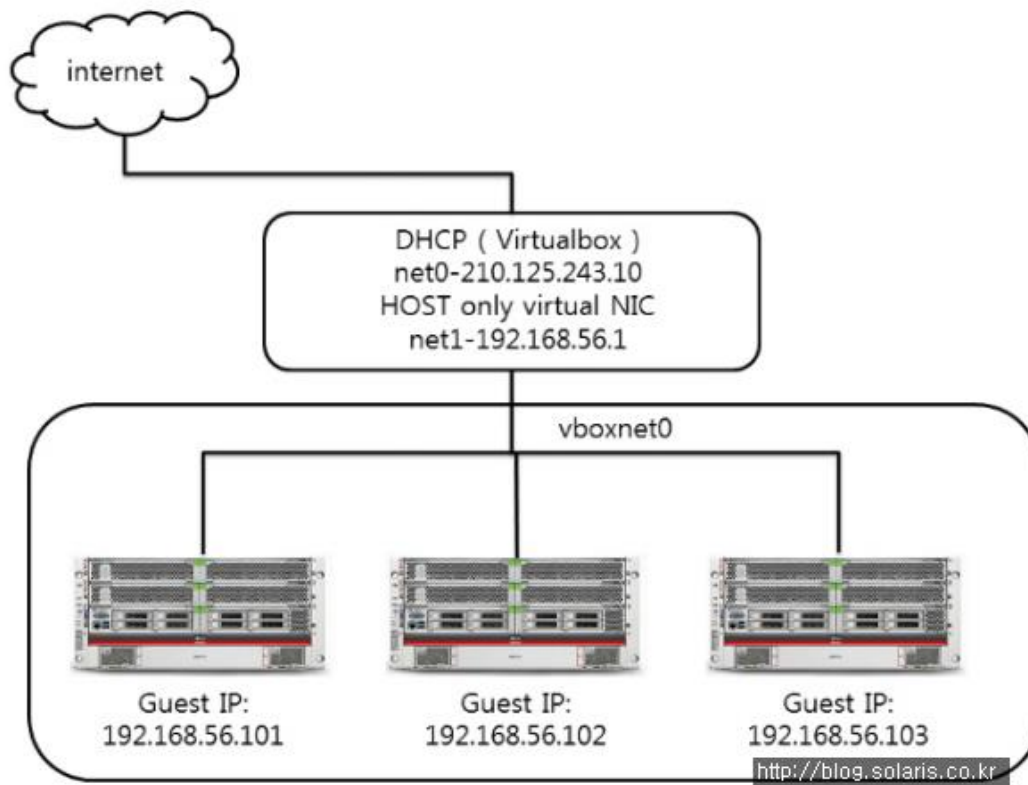
Internal Networking



- 내부 네트워크가 별도의 vboxnet0로 격리된 네트워크로 동작함
- 다수의 내부 가상 시스템과 통신을 지원하고 외부로부터 격리하여 보안을 강화하도록 조치 가능
- 호스트 네트워크와 분리되어 동작하여 인터넷 연결 지원은 되지 않음
- 필요에 따라 vboxnet1, vboxnet2 등 가상 네트워크 추가 가능

Virtualbox Network 연결 방식

▪ Host Only Networking



- 내부 네트워크 연결 방식과 비슷하게 연결되지만 호스트와의 통신이 지원되고 DHCP를 이용한 네트워크 IP 할당이 가능함
- 기본 네트워크 대역은 192.168.56.0/24 대역이며 필요시 변경 가능
- 호스트는 인터넷 연결과 가상 머신들과 연결이 가능하지만, 가상 머신들은 호스트와 네트워킹은 지원되지만 인터넷은 불가능함.

※ Host Only Networking Adapter 설치가 안된 경우 Windows CMD 창을 연후 아래 명령어를 입력하면 된다

C:\Program Files\Oracle\VirtualBox> VBoxManage.exe hostonlyif create

Linux 부팅 및 종료

■ 부팅 및 종료 명령어

```
$ reboot  
$ reboot -f
```

시스템 재부팅
시스템 강제 재부팅

```
$ halt  
$ halt -f
```

시스템 종료
시스템 강제 종료

```
$ shutdown -h now  
$ shutdown -h 10  
$ shutdown -h 15:10  
$ shutdown -r now  
$ shutdown -c
```

지금 바로 시스템 종료
10분 후에 시스템 종료
15:10분에 시스템 종료
지금 바로 시스템 재부팅
시스템 종료 예약 취소

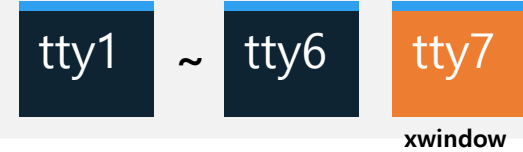
```
$ init 0  
$ init 1  
$ init 2  
$ init 3  
$ init 4  
$ init 5  
$ init 6
```

Runlevel0 시스템 종료 모드
Runlevel1 단일 사용자 모드
Runlevel2 NFS가 없는 다중 사용자 모드
Runlevel3 CLI 의 다중 사용자 모드
Runlevel4 예비 Level
Runlevel5 xwindows 다중 사용자 모드
Runlevel6 재부팅 모드

자주 사용하는 유용한 팁

1

부팅 후 Alt + 1 ~ 6 번까지 CLI 터미널을 이동하며 Multi Tasking 가능하다.



2

명령어 입력시 일부분만 입력 후 Tab을 누르면 자동 완성 기능을 사용할 수 있다.



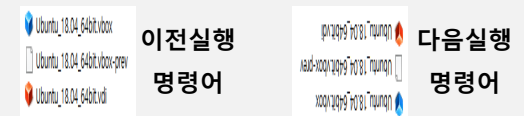
3

history 명령어를 통해 이전에 입력한 명령어 기록을 확인할 수 있다.



4

위 또는 아래 화살표 입력 버튼을 통해 기존 입력 명령어를 불러올 수 있다.



5

Man 명령어를 통해 Linux 명령어의 사용자 매뉴얼을 확인 할 수 있다.



6

Ctrl+C 또는 Ctrl+Pause Break 버튼으로 실행중인 작업을 취소 또는 종료 할 수 있다.



자주 사용하는 유용한 팁

▪ history 사용예

\$ history	: 기존 입력 명령어를 화면에 출력
\$ history <number>	: 입력한 수 만큼의 명령어 출력
\$ history -c	: 기존 입력 명령어 모두 삭제
\$ history -w <file name>	: 기존 입력 명령어를 파일로 저장

▪ man 사용예

\$ man <command> /: 검색, n: 다음검색, N: 이전검색 q:종료	: 입력된 명령어의 매뉴얼 열기
\$ man -k <command>	: 간략한 명령어 설명 (apropos 검색 결과를 출력)
\$ <command> --help	: 간단한 명령어 사용법 출력

자주 사용하는 유용한 팁

▪ locate 사용예 – 성능은 뛰어나지만 실시간 반영이 아님

\$ sudo updated	Locate DB 업데이트, cron에 등록되어 매일 새벽에 자동 실행됨
\$ locate <file-name>	해당 문자열이 포함된 파일의 위치출력
\$ locate -n 10 <file-name>	해당 문자열이 포함된 10개 파일 위치 출력

▪ find 사용예 – 검색 조건에 따라 성능은 떨어지지만 실시간 반영

\$ find <path> <option> <file name>	Find 명령어 사용법
\$ find -name '*.py'	현재 하위 디렉토리 py확장 파일 검색
\$ find /usr/share -name 'gnome'	gnome 파일 검색
\$ find /usr/share -name 'gnome*'	gnome 로 시작하는 모든 파일 검색
\$ find /usr/share -name '*gnome' -ls	gnome로 끝나는 모든 파일 검색하여 ls 명령어 형태로 화면에 출력
\$ find /usr/share -name 'con*' -type d	con으로 시작하는 디렉토리 검색
\$ find ./ -ctime +10	최종갱신일 10일 이상된 파일검색(+/-)
\$ find ./ -ctime 10	최종갱신일 10일인 파일검색
# atime(최종접근시간), mtime(최종변경시간), ctime(이름,퍼미션,이동 등의 최종변경시간)	

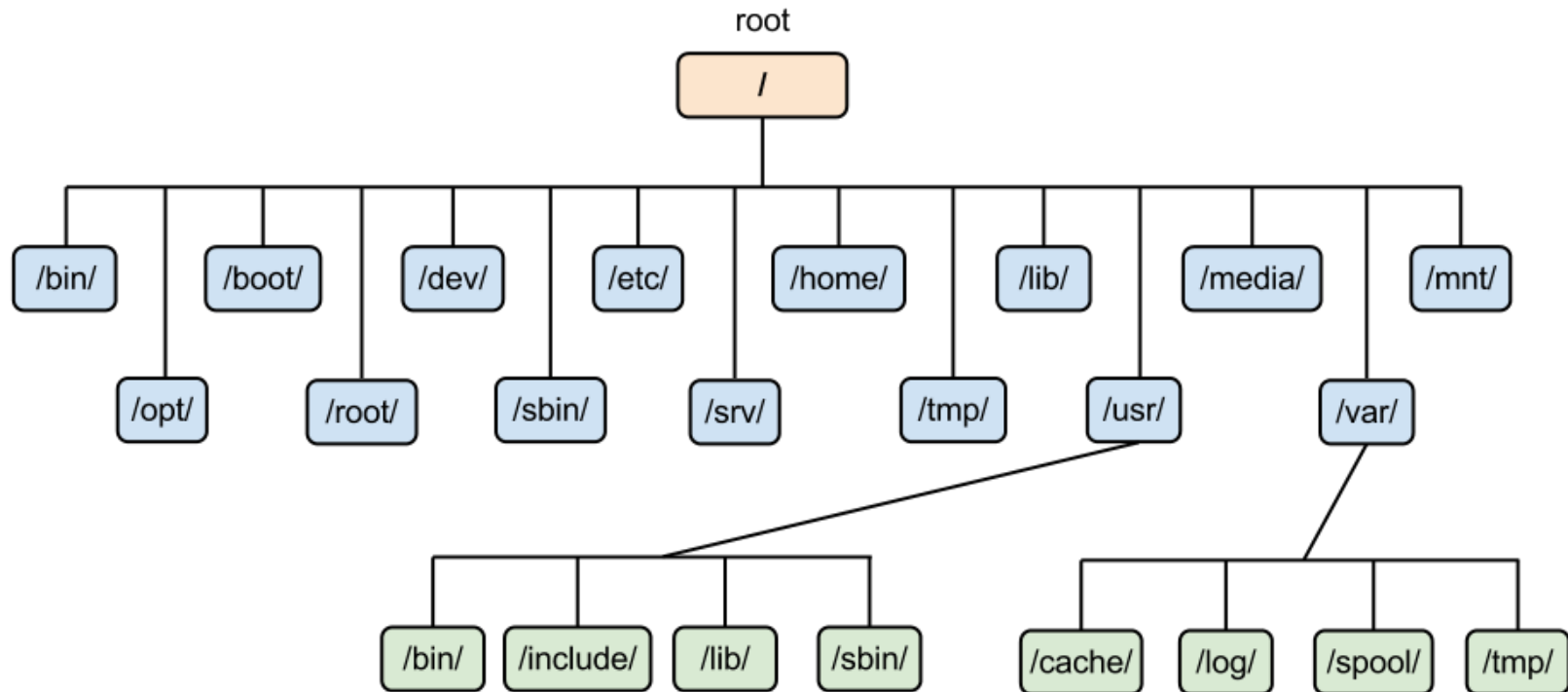
3. File 및 Directory 관리 명령어

- 1) Linux 디렉토리 구조
- 2) 기본 명령어
- 3) vi 편집기
- 4) pipe, filter, redirection

Linux 디렉토리 구조

Linux 디렉토리 구조

- 사용 용도별 각 파일이 디렉토리에 나누어져 설치됨



Linux 디렉토리 구조

▪ root & home

- home 디렉토리는 리눅스 사용자들의 개인 공간
- 보통 일반 사용자는 home 밑에 별도의 설정이 없는 경우 자신의 아이디로 된 디렉토리가 배정됨
=> 로그인 시 처음 작업 디렉토리가 됨
- /root는 root 의 홈 디렉토리

▪ bin &/sbin

- 리눅스 사용에 필수적인 명령어들을 모아놓은 디렉토리
- sbin 은 시스템 관리를 위한 명령들 => root에게만 허용됨
- 이렇게 명령어들을 모아놓은 디렉토리의 위치는 PATH 환경변수에 저장되어 있음
=> 셸에서 \$echo \$PATH 를 실행하면 현재 PATH경로 확인 가능

▪ lib

- 리눅스 상에서 자주 사용되는 라이브러리를 모아놓음
- Shared object file format(.so)
- 윈도우에서의 Dynamic Linked Library(DLL)을 모아놓은 폴더로 생각할 수 있음

Linux 디렉토리 구조

▪ etc

- 리눅스 내의 설정 파일들이 위치함
- 주요 설정 파일들
 - motd : 로그인 성공 시 처음에 띄워주는 문구
 - passwd : 리눅스 사용자에게 대한 정보 (예전에는 사용자의 비밀번호(!) 도 있었음)
 - shadow : 리눅스 사용자 패스워드 정보 (root만 볼수 있음)
 - sudoers : sudo 권한을 가진 사용자와 권한에 대한 정보
 - fstab : 파일 시스템과 마운트 포인터
 - hosts : 도메인 네임
 - issue : 리눅스 배포판 및 버전
 - skel/ : 새로운 사용자가 추가될 때 홈 디렉토리에 기본적으로 복사해줄 파일을 포함한 디렉토리
 - init.d/ : 컴퓨터가 부팅될 때 실행될 파일들을 넣어둔 디렉토리
- ※ 이 외에도 많은 파일들이 중요한 설정을 가지고 있음

▪ proc

- RAM 위에 존재하며 커널에 의해 제어됨
- 현재 실행되는 프로세스와 실제로 사용되는 장치, 커널이 수집한 하드웨어 정보가 저장됨
- 사용자가 /proc이나 하위 파일에 접근할 때마다 커널에서 파일 내용을 동적으로 만들어냄
- 각 프로세스는 고유의 ID를 가지고 있으며 이 아이디를 가진 디렉토리 밑에 각종 정보를 저장=>

- cmdline : 프로세스를 시작한 명령행 내용	/proc/meminfo : 총 메모리 사용 현황
- status : 프로세스의 내부 상태 정보	/proc/stat : 시스템의 상태에 관한 정보
- cwd : 프로세스의 현재 작업 디렉토리	/proc/uptime : 시스템이 부팅된 후 흐른 시간
	/proc/version : 현재 실행되는 커널 버전

Linux 디렉토리 구조

▪ dev

- 실제로 하드 디스크에 존재하지는 않는 정보를 저장함
- 모든 하드웨어(hdd, cdrom, 마우스, 가상 장치, etc...) 를 파일로 인식하며 이를 장치 파일이라 함
 - psaux PS/2 마우스 장치
 - tty : teletypewriter, 사용자와 시스템 사이를 중계하는 역할
사용자와 시스템 사이의 입출력은 모두 여기를 통해 이루어짐
서버 시스템에 직접 연결된 장치에서 사용함
 - pts : telnet, ssh 등을 이용해 원격으로 접속할 경우
(pseudo TTY slave) 유사 장치 : 실제 장치와 관련이 없음
 - null : 이 장치로 들어간 데이터는 모두 사라짐
 - zero : 이 장치에서는 항상 null을 반환

▪ mnt & media

- 둘 다 외부 저장 매체가 마운트 되는 디렉토리
- 마운트 : 한 파일 시스템의 루트 디렉토리를 다른 디렉토리에 붙임으로써 디렉토리를 사용할 수 있게 만들어주며, 모든 파일 시스템들을 마치 그들이 속해있는 파일시스템의 서브 디렉토리인 것처럼 사용 가능하게 만드는 것
- 문서에는 media 는 주로 플로피 디스크, CD 등이 마운트 되고 mnt 는 수동적으로 마운트 시켜야 하는 저장 매체를 마운트 되는 곳으로 쓰도록 되어있다.
- 하지만, 이제는 많은 저장 매체들이 자동으로 마운트 되기 때문에 주로 /media(특히 우분투) 를 사용하는 경우가 많지만 mnt를 사용하는 것은 자유

Linux 디렉토리 구조

▪ tmp

- 리눅스 상에서 실행되는 프로세스들의 임시 파일들을 저장하는 위치
- 종료 시 이 위치에 있는 파일들은 모두 삭제됨

▪ boot

- 부팅에 필요한 필수 파일들이 저장됨
- lilo, grub 등의 부트 로더와 커널이 위치함
 - ※ 부트 로더 : 컴퓨터를 켤 때 가장 먼저 실행되는 프로그램
 - => OS의 커널을 로드하고 몇몇 커널 파라미터를 커널에 넘김

▪ var

- 리눅스 상에서 자주 변경되는 데이터들을 모아놓음

/var/lib: 일반적인 시스템 운용시 계속 갱신되는 파일들을 위한 공간

/var/local: /usr/local 아래에 설치된 프로그램들의 다양한 데이터가 보관

/var/lock : 잠금 파일(lock file)이 있는 곳이다.

/var/log : 다양한 프로그램들의 로그 파일

/var/log/wtmp : 시스템의 모든 로그인, 로그아웃 정보를 기록

/var/log/messages: 커널과 시스템 프로그램들의 모든 메시지

/var/run : 시스템의 현재 정보, 부팅 시 리셋

/var/run/utmp : 현재 로그인한 사용자들에 대한 정보

/var/spool : 대기 상태에 있는 작업들을 위한 디렉토리

/var/tmp : /tmp에 있는 임시 파일들보다는 좀 더 오래 유지될 필요가 있는 임시 파일들, 부팅 시 지워지지 않는다.

Linux 디렉토리 구조

■ usr

- 리눅스 상에서 가장 큰 공간을 사용하는 디렉토리이며, 리눅스 배포판에 따라 많은 차이를 보임
- 어플리케이션 수준의 프로그램들이 여기에 있음
- usr 밑의 bin, lib,... 등은 사용자가 자유롭게 다룰 수 있으며 루트 디렉토리의 같은 이름을 가진 디렉토리와의 같은 역할을 한다. (대신 우선순위가 떨어진다.)

usr/bin : 실행 가능한 명령어, 주로 콘솔이나 X에서 사용되는 명령어들

usr/sbin : 시스템 관리를 위한 명령어, 주로 서버 관리용

usr/lib : 프로그램과 하위 시스템을 위한 라이브러리

/usr/man, /usr/info, /usr/doc : 각각 매뉴얼 페이지, GNU Info 문서들, 그리고 기타 다른 문서

usr/include : C 를 위한 헤더 파일

usr/X11R6 : X window

usr/src : 시스템에 빌드하는 프로그램의 소스

usr/share : 읽기 전용 자료, 주로 매뉴얼이나 문서 자료
usr/local : 리눅스에서 필수적인 파일이나 배포판에서 확장되는 파일들을 제외한 응용 프로그램 파일들을 저장
응용 프로그램 설치 시 프로그램이 독자적으로 명령어를 추가하거나 라이브러리가 추가적으로 필요로 할 경우 등 추가적인 설정이 필요할 때, 이 위치에 저장하기 때문에 usr와 흡사한 구조로 되어있다.

/bin : 문서에 명시되어 있는 명령어 및 필수적인 명령어

/usr/bin : 주로 콘솔과 X에서 사용되는 명령어

/usr/local/bin : 그 외 어플리케이션에 의해 추가되는 명령어

sbin, lib, etc, 등도 이와 비슷한 구조를 가짐

■ opt

- 주로 규모가 있는 소프트웨어 패키지를 저장. gnome, kde, 오픈오피스, 파이어폭스 등등 ...

기본 명령어

■ 파일 및 디렉토리 관리를 위한 기본 리눅스 명령어 (가장 기본이며 필수)

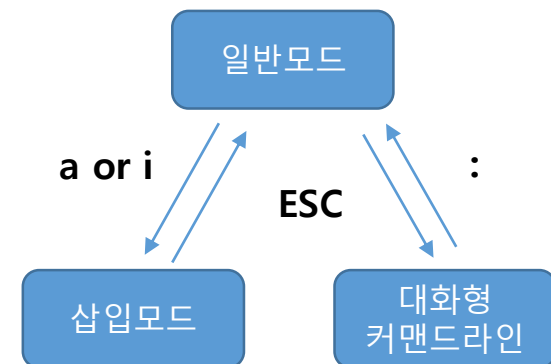
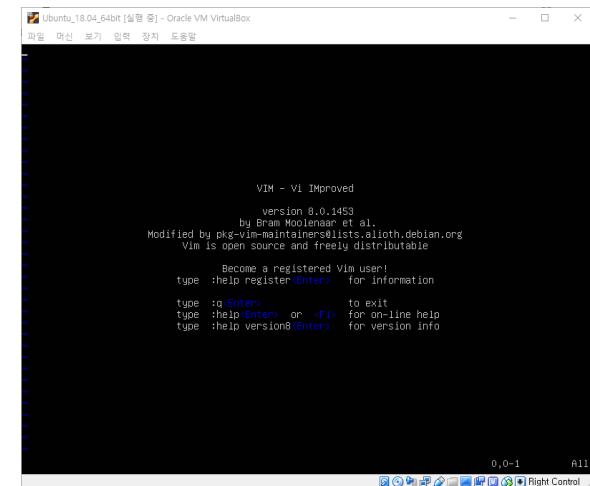
명령어	명령어 구분	설명
pwd	bash 셸 외부 명령	현재 디렉토리를 표시한다.
cd	bash 내장 명령	현재 작업 디렉토리를 지정된 디렉토리로 바꾼다.
ls	bash 셸 외부 명령	디렉토리 내용의 목록을 표시한다
clear	bash 셸 외부 명령	터미널 에뮬레이터 또는 가상 콘솔 터미널에서 텍스트를 지운다
cp	bash 셸 외부 명령	지정된 파일을 다른 위치에 복사한다
mv	bash 셸 외부 명령	지정된 파일을 다른 위치로 이동하거나 파일 이름을 바꾼다
touch	bash 셸 외부 명령	새파일을 만들거나 기존 파일의 타임스탬프를 업데이트한다
rm	bash 셸 외부 명령	지정된 파일을 삭제한다
cat	bash 셸 외부 명령	지정된 파일의 내용을 표시한다
who	bash 셸 외부 명령	현재 시스템에 로그인한 사용자를 표시한다
mkdir	bash 셸 외부 명령	현재 디렉토리에 지정된 디렉토리를 만든다
rmdir	bash 셸 외부 명령	지정된 디렉토리를 삭제한다
ln	bash 셸 외부 명령	지정된 파일에 대한 심볼릭 링크 나 하드 링크를 만든다
alias	bash 내장 명령	지정된 명령에 대한 별명을 정의한다
which	bash 셸 외부 명령	특정명령어의 위치를 찾아준다.

vi 편집기

- Ubuntu 에서는 vi 명령을 입력하면 vim(vi improved) 가 실행됨
- vim은 GNU 프로젝트 오픈소스로 vi 편집기를 이식할 때 기능을 개선함

vim 편집 명령

명령	설명
x	현재 커서 위치에 있는 문자를 지운다
dd	현재 커서 위치에 있는 줄을 지운다
dw	현재 커서 위치에 있는 단어를 지운다
d\$	현재 커서 위치에서 줄 끝까지를 지운다
J	현재 커서 위치에서 줄의 끝에 있는 줄바꿈을 지운다 (두 행을 합친다)
u	이전 편집 명령을 취소한다
a	현재 커서 위치 뒤에 데이터를 추가한다
A	현재 커서 위치의 줄 끝에 데이터를 추가한다.



vi 편집기

대화형 커맨드 라인 명령

명령	설명
q	버퍼 데이터에 아무런 변경도 이루어지지 않은 경우 종료
q!	버퍼 데이터에 대한 모든 변경사항을 취소하고 종료
wq	파일에 버퍼의 데이터를 저장하고 종료
w filename	파일을 filename으로 지정된 다른 파일에 저장

찾기 및 바꾸기(일반 모드)

명령	설명
/	커서가 메시지 라인으로 옮겨가고 찾으려는 텍스트를 입력하면 입력한 텍스트를 찾는다 n → 다음단어 찾기, N → 이전단어 찾기
:s/fi/foo/g	한 줄에서 나타나는 모든 fi를 foo로 바꾼다
:n,ms/fi/foo/g	n번째 줄과 m번째 줄 사이에 나타나는 모든 fi를 foo로 바꾼다
:%s/fi/foo/g	전체 파일에서 나타나는 fi를 foo로 바꾼다
:%s/fi/foo/gc	전체 파일에서 나타나는 fi를 건마다 사용자의 확인을 받아 foo로 바꿀지 바꾸지 않을지 결정함

pipe, filter, redirection

■ 파이프(pipe)

- 두 프로그램을 연결해주는 연결 통로의 의미
- " | " 문자를 사용함 (₩키를 Shift와 함께 누른 글자)

```
$ ls -l /etc | more
```

```
$ ls -l /etc | ls -l /dev | more
```

```
$ ls -l /etc | less
```

etc 디렉토리 파일을 출력하고 화면 페이지가 넘어가면 키 입력에 따라 출력

■ 필터(filter)

- 필요한 것만 걸러주는 명령어로 grep, tail, wc, sort, awk, sed 등 주로 pipe와 같이 사용됨

```
$ cat .profile | grep package
```

package 단어 포함 라인만 출력

```
$ cat .profile | tail -n 10
```

마지막 10줄의 내용을 출력

```
$ cat .profile | wc -l
```

전체 라인 count 출력

```
$ cat .profile | sort -r
```

내림차순으로 파일 내용을 출력

pipe, filter, redirection

리다이렉션(redirection)

- 표준 입출력의 방향을 바꿔줌
- 표준 입력은 키보드, 표준 출력은 모니터이지만 이를 파일로 처리하고 싶을때 사용

리다이렉션 기호	방향	설명
>	표준 출력	명령 > 파일 : 명령의 결과를 파일로 저장
>>	표준 출력 (Append)	명령 >> 파일 : 명령의 결과를 기존 파일 데이터에 추가
<	표준 입력	명령 < 파일 : 파일의 데이터를 명령에 입력

```
$ ls -l > aa.txt
```

명령의 결과를 aa.txt에 씀, 기존 파일
이 있을 경우 overwrite

```
$ ls -l >> aa.txt
```

명령의 결과를 aa.txt에 추가(append)

```
$ sort < aa.txt
```

aa.txt 파일을 정렬해서 화면에 출력

```
$ sort < aa.txt > bb.txt
```

aa.txt 파일을 정렬 후 bb.txt에 씀

compress 및 decompress

■ 파일 아카이브

명령	설명
tar	<ul style="list-style-type: none"> ▪ 대상 파일을 묶어서 하나로 만든다. (압축은 이뤄지지 않음), ▪ tar(tape archive)는 여러 파일이나 디렉토리를 묶어서 만든 마그네틱 테이프와 같은 이동식 저장 장치에 보관하기 위해 사용하는 명령이었음 ▪ 백업 및 복원을 위한 용도로 사용되어짐
	<div> <div> c – 새로운 tar 파일의 내용을 출력 t – tar 파일의 내용을 출력 x – tar 파일에서 원본 파일을 추출 r – 새로운 파일을 추가 u – 수정된 파일을 업데이트 </div> <div> f – 아카이브 파일이름을 지정 v – 처리하고 있는 파일 정보 출력 h – 심볼릭 링크의 원본 파일을 포함 p – 파일 복구시 원래 접근 권한 유지 z – gzip로 압축하거나 해제 </div> </div>

```
$ touch test1 test2 test3
$ tar cvf test.tar test*
$ tar tvf test.tar
$ rm test1 test2 test3
$ tar xvf test.tar
$ tar rvf test.tar <filename>
```

아카이브 테스트 새파일 3개를 생성
아카이브 만들기
아카이브 내용보기
아카이브 테스트 파일 삭제
아카이브 풀기
아카이브 파일 추가하기

compress 및 decompress

■ 파일 압축 및 아카이브

```
$ gzip test.tar
$ gunzip test.tar.gz
$ tar cvfz test.tar.gz test*
$ tar xvfz test.tar.gz
```

```
$ zip test.zip ./*
$ zip test.zip -r ./*
```

아카이브 압축하기
아카이브 압축 해제
아카이브 및 압축 실행
아카이브 및 압축 해제

하위폴더 모두 zip 압축
하위폴더로 zip 압축 해제

<https://archive.apache.org/dist/>

인터넷에서 파일 다운로드

```
$ wget <url/filename>
```

```
$ wget https://archive.apache.org
/dist /hadoop/hadoop-2.8.0.tar.gz
```

Apache Software Foundation Distribution Directory

The directories linked below contain current software releases from the Apache Software Foundation projects. Older non-recommended releases can be found on our [archive site](#).

To find the right download for a particular project, you should start at the project's own webpage or on our [project resource listing](#) rather than browsing the links below.

Please do not download from apache.org! If you are currently at apache.org and would like to browse, please visit [a nearby mirror site](#) instead.

Projects

Name	Last modified	Size	Description
 Parent Directory		-	
 META/	2018-05-20 06:39	-	
 abdera/	2017-10-04 10:56	-	
 accumulo/	2018-05-15 19:58	-	
 ace/	2017-10-04 11:11	-	
 activemq/	2018-05-21 12:37	-	
 airavata/	2018-05-04 20:46	-	
 allura/	2018-05-04 15:14	-	
 ambari/	2018-05-04 21:08	-	

4. User 및 Permission 관리 명령어

- 1) Linux User 추가/제거
- 2) Linux Group 사용하기
- 3) 권한 및 소유권
- 4) 권한 및 소유권 변경하기

Linux User 추가/제거

▪ su 와 sudo 명령어

- su : 로그아웃을 하지 않고 다른 사용자의 계정으로 전환
- sudo : super user 의 보안권한으로 프로그램을 구동할 수 있음

sudo 권한으로 명령어를 실행하려면 root 계정 암호를 입력해야함

모든 계정이 sudo 명령어를 이용해 super user의 권한을 사용할 수 있지는 않음

```
$ cat /etc/passwd
```

오직 super user만이 파일 접근 가능

```
$ sudo cat /etc/passwd
```

super user 권한을 빌려 파일 접근

▪ 사용자 추가 (사용자 추가시 /etc/skel 의 파일을 home 디렉토리로 복사함)

```
$ sudo useradd -m ubuntu2
```

home 디렉토리 생성하여 사용자 추가
- adduser 명령어도 사용가능

```
$ sudo passwd ubuntu2
```

새 사용자의 암호 설정

```
$ su ubuntu2
```

새 사용자로 계정 전환(su - 새 사용자의 환경변수를 적용하여 계정 전환)

```
$ sudo userdel -r ubuntu2
```

home 디렉토리 포함하여 사용자 삭제

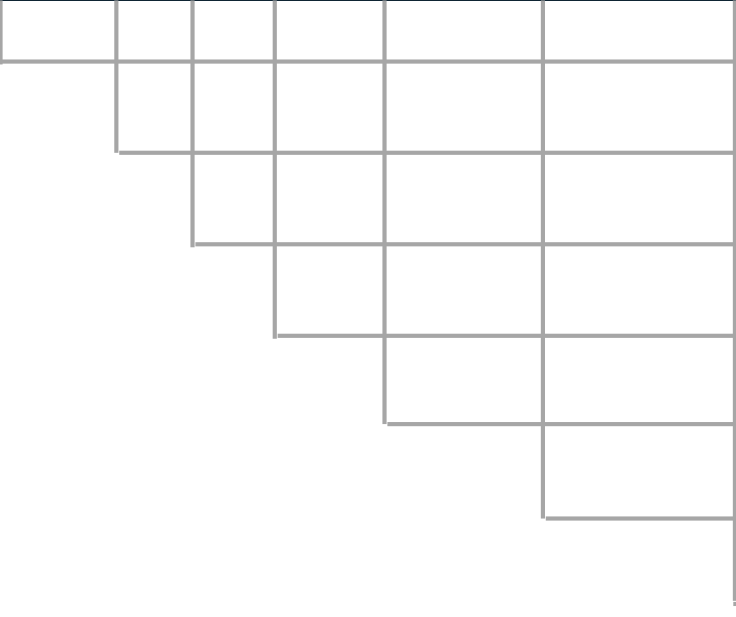
Linux User 추가/제거

Linux 보안

- 개별 사용자 및 그룹이 각 파일 및 디렉토리에 대한 일련의 보안 설정에 따라 파일에 접근 가능
- 사용자 권한은 계정이 만들어질 때 할당되는 UID(사용자 ID)로 추적 관리됨

/etc/passwd (사용자에 대한 여러 정보를 저장)

```
$ cat /etc/passwd  
root : x : 0 : 0 : root : /root : /bin/bash
```

- 
- 로그인 사용자의 이름
 - 사용자의 암호(/etc/shadow 에 별도 저장됨)
 - 사용자 계정의 숫자 UID
 - 사용자 계정의 숫자 그룹 ID(GID)
 - 사용자 계정의 설명 텍스트(주석 필드)
 - 사용자의 HOME 디렉토리의 위치
 - 사용자의 기본 셸

Linux User 추가/제거

▪ /etc/shadow (암호 및 암호 관리 파일)

- 사용자 암호를 더 정밀하게 제어 가능
- 암호에 대한 총 아홉 가지 설정 가능

```
$ sudo cat /etc/shadow
ubuntu : $2309jdfkSDAGg : 11627 : 0 : 99999 : 7 :::
```

- 로그인 사용자 이름
- 암호화된 암호
- 암호가 마지막으로 변경된 날짜(1970.1.1 기준)
- 암호 변경 가능 최소 일수
- 암호 변경까지 남은 일 수
- 사용자 암호 변경 경고 메시지 남은 일수
- 계정 말료되어 비활성화 까지 남은 일수
- 사용자 계정 비활성화 된 날(1970.1.1 기준)
- 향후 사용을 위해 유보된 필드

Binary : 01111111 11111111 11111111 11111111

Decimal : 2147483647

Date : 2038-01-19 03:14:07 (UTC)

Date : 2038-01-19 03:14:07 (UTC)

Binary : 10000000 00000000 00000000 00000000

Decimal : -2147483648

Date : 1901-12-13 20:45:52 (UTC)

Date : 2038-01-19 03:14:08 (UTC)

Linux User 추가/제거

▪ /etc/shadow (암호 및 암호 관리 파일)

- 사용자 암호를 더 정밀하게 제어 가능
- 암호에 대한 총 아홉 가지 설정 가능

```
$ sudo cat /etc/shadow
ubuntu : $2309jdfkSDAGg : 11627 : 0 : 99999 : 7 :::
```

- 로그인 사용자 이름
- 암호화된 암호
- 암호가 마지막으로 변경된 날짜(1970.1.1 기준)
- 암호 변경 가능 최소 일수
- 암호 변경까지 남은 일 수
- 사용자 암호 변경 경고 메시지 남은 일수
- 계정 말료되어 비활성화 까지 남은 일수
- 사용자 계정 비활성화 된 날(1970.1.1 기준)
- 향후 사용을 위해 유보된 필드

Binary : 01111111 11111111 11111111 11111111

Decimal : 2147483647

Date : 2038-01-19 03:14:07 (UTC)

Date : 2038-01-19 03:14:07 (UTC)

Binary : 10000000 00000000 00000000 00000000

Decimal : -2147483648

Date : 1901-12-13 20:45:52 (UTC)

Date : 2038-01-19 03:14:08 (UTC)

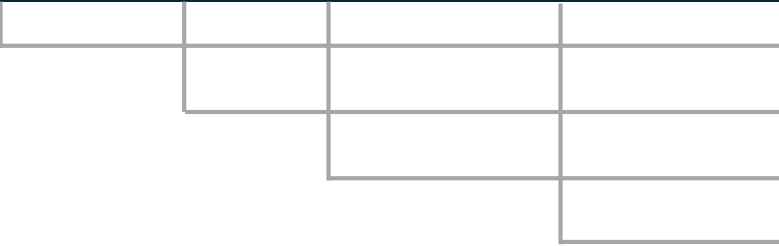
Linux User 추가/제거

▪ /etc/sudoers

- sudo 명령을 사용하여 root 권한으로 명령을 수행할 수 있도록 함

```
# User privilege specification
```

```
root    ALL=(ALL)    ALL
```

- 
- 사용자 이름
 - sudo 사용 가능한 터미널 (모든 터미널에서 사용)
 - 권한을 가질 수 있는 사용자 (모든 사용자의 권한)
 - 실행할 수 있는 명령어 (모든 명령어를 사용)

```
# Host alias specification
```

```
Host_Alias UNET = 10.1.2.0/255.255.255.0
```

```
# User alias specification
```

```
User_Alias UADMIN = ubuntu2, ubuntu3, ubuntu4
```

```
# Cmnd alias specification
```

```
Cmnd_Alias PAC = /bin/chown, /usr/bin/apt-get
```

Linux User 추가/제거

■ 사용자 계정 수정 유틸리티

- 사용자 계정의 정보를 변경하기 위한 특정한 기능을 제공

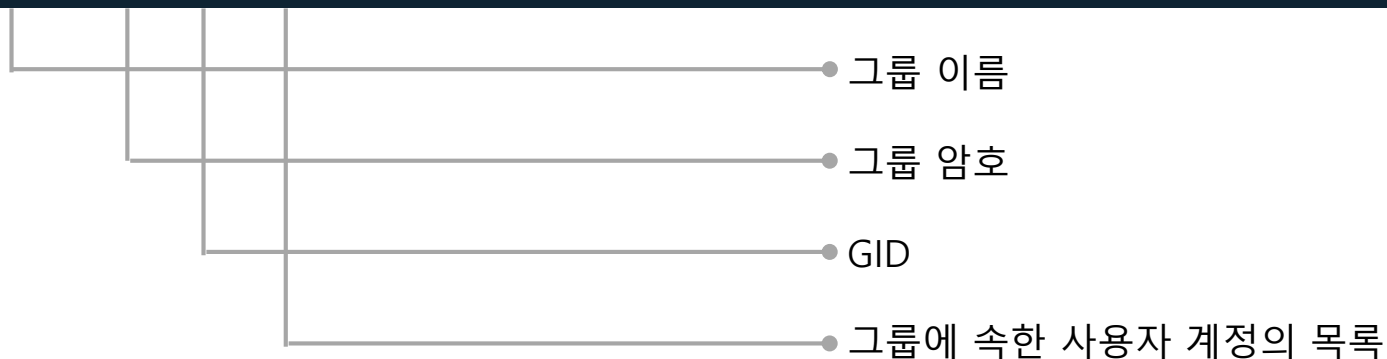
명령	설명
usermod	<ul style="list-style-type: none"> ■ 사용자 계정 필드를 편집하는 것은 물론 기본 및 보조 그룹 구성원을 지정 ■ -l (로그인 이름을 변경), -L (계정 잠금), -p (계정 암호 변경), -U (계정 잠금 해제) ■ \$ usermod <사용자 계정> -l <변경할 사용자 계정>
passwd	<ul style="list-style-type: none"> ■ 기존 사용자의 암호를 변경 ■ \$ sudo passwd <사용자 계정>
chpasswd	<ul style="list-style-type: none"> ■ 로그인 이름과 암호가 한 쌍으로 된 파일을 읽어 암호를 갱신 ■ \$ sudo chpasswd < users.txt : userid:passwd 쌍으로 내용이 작성되어 있어야함
chfn	<ul style="list-style-type: none"> ■ 사용자 계정의 주석 정보 변경 ■ \$ chfn
chsh	<ul style="list-style-type: none"> ■ 사용자 계정의 기본 셸 변경 ■ \$ chsh

Linux Group 사용하기

▪ /etc/group

- 시스템이 사용하는 각 그룹에 대한 정보가 포함됨,

```
$ cat /etc/group  
root : x : 0 :
```



▪ 그룹 추가/수정

<code>\$ sudo groupadd ubuntgro</code>	ubuntgro 그룹을 생성
<code>\$ sudo groupmod -n gubuntu ubuntgro</code>	그룹이름 변경, -g GID 변경
<code>\$ sudo groupdel gubuntu</code>	ubuntgro 그룹을 삭제

권한 및 소유권

파일의 권한

```
ubuntu@ubuntu:~$ ls -al
total 48
drwxr-xr-x 5 ubuntu ubuntu 4096 Jun 26 14:52 .
drwxr-xr-x 5 root root 4096 Jun 26 13:50 ..
-rw-r--r-- 1 ubuntu ubuntu 3580 Jun 26 13:28 .bash_history
-rw-r--r-- 1 ubuntu ubuntu 220 Apr 4 18:30 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Apr 4 18:30 .bashrc
drwx--- 2 ubuntu ubuntu 4096 Jun 22 15:50 .cache
drwx--- 3 ubuntu ubuntu 4096 Jun 22 16:06 .config
drwx--- 3 ubuntu ubuntu 4096 Jun 22 15:50 .gnupg
-rw-r--r-- 1 ubuntu ubuntu 34 Jun 23 17:07 .lessht
-rw-r--r-- 1 ubuntu ubuntu 807 Apr 4 18:30 .profile
-rw-r--r-- 1 ubuntu ubuntu 0 Jun 22 15:59 .sudo_as_admin_successful
-rw-r--r-- 1 root root 6271 Jun 26 13:39 .viminfo
ubuntu@ubuntu:~$ _
```

파일소유자와 그룹

- 모든 다른 사람들의 권한
- 그룹 구성원의 권한
- 파일 소유자의 권한
- 개체 유형 정의 문자

유형 정의 문자

- 파일
d 디렉토리
l 링크

엑세스 권한

r : 읽기 권한
w : 쓰기 권한
x : 실행 권한

권한 및 소유권

■ 권한 코드와 umask

권한	2진수	8진수	설명
---	000	0	■ 권한없음
--x	001	1	■ 실행 전용 권한
-w-	010	2	■ 쓰기 전용 권한
-wx	011	3	■ 쓰기 및 실행 권한
r--	100	4	■ 읽기 전용 권한
r-x	101	5	■ 읽기 및 실행 권한
rw-	110	6	■ 읽기 및 쓰기 권한
rwX	111	7	■ 읽기, 쓰기 및 실행 권한

- umask 명령은 사용자가 만든 파일이나 디렉토리에 대한 기본 권한을 설정함.
- umask 값은 마스크 값으로 보안 수준에 대하여 주고 싶지 않은 권한을 마스크 하는 것
- 파일의 모든 권한인 666(읽기/쓰기)에서 umask 값을 뺀 값이 실제 권한이 됨
- 예를 들어 umask 값이 222일때 새로 생성되는 파일은 쓰기 권한을 제외한 444(읽기) 권한만을 가지게 됨

권한 및 소유권 변경하기

▪ chmod와 chown을 이용한 권한 및 소유권 변경

chmod 사용 예제

```
$ touch test && ls -al test
```

```
-rw-rw-r-- 1 ubuntu ubuntu 0 Jun 26 15:20 test
```

test 파일 생성 및 권한 확인

```
$ chmod 760 test
```

```
-rwxrw---- 1 ubuntu ubuntu 0 Jun 26 15:20 test
```

8진수를 이용한 권한 변경

```
$ chmod u-x test
```

```
-rw-rw---- 1 ubuntu ubuntu 0 Jun 26 15:20 test
```

기호를 이용한 권한 변경
(u-사용자, g-그룹,
o-다른 사용자, a-모든 사용자)

chown 사용 예제

```
$ chown ubuntu2 test
```

```
$ chown ubuntu2.ubuntu2 test
```

```
$ chown .ubuntu test
```

```
$ chown ubuntu2. test
```

사용자 소유권 변경
그룹 및 사용자 소유권 변경
그룹 소유권 변경
그룹 및 사용자 소유권 변경
(사용자, 그룹 이름 일치 시)

5. Network 관리 명령어

- 1) 네트워크 관리 명령어
- 2) 네트워크 관리 방법
- 3) 방화벽 설정

네트워크 관리 명령어

■ 네트워크 관리 파일 및 명령어

파일 및 명령어	설명
/etc/hostname	ubuntu linux의 호스트 이름을 저장하는 파일
/etc/hosts	Linux OS 가 호스트 이름을 IP 주소에 매핑할 때 사용하는 파일
/etc/interface	네트워크 설정 정보를 등록하는 파일
/etc/recv.conf	DNS 서버를 관리하는 파일
ifconfig	Linux에서 네트워크 아답터 및 네트워크 설정 정보를 출력
ping	IP 프로토콜의 기본 명령어로 상대 컴퓨터와 연결이 되는지 확인
ssh	안전한 원격 통신을 위해 사용
netstat	네트워크 모니터링 도구. 서비스포트 활동상태를 보여줌
nslookup	도메인 이름과 IP 주소를 확인하는 기능을 가진 네트워킹 관리 툴
telnet	리눅스 서버 원격접속에 가장 대표적으로 사용되는 명령어
tracert	네트워킹 관리 도구로 IP 경로를 추적하는 명령어
scp	서버간 파일 전송 및 수신을 위한 명령어
nmap	hostname 또는 IP address 를 통해 현재 열려있는 서비스 포트를 검색
ifup, ifdown	ethx 이더넷 네트워크를 작동 또는 중지 (apt 에서 설치 필요)

네트워크 설정 방법

■ 네트워크 관리 필수 명령어

\$ ifconfig	네트워크 정보 관리
\$ ping	네트워크 연결 확인
# ssh 설정정보 /etc/ssh/sshd_config	
\$ ssh localhost	localhost로 연결
\$ ssh ubuntu@10.0.2.16	IP의 ubuntu계정 연결
\$ ssh ubuntu@ubuntu2.com -p1022	1022포트로 연결 시도
# scp <file> username@server:<path>	파일 전송 명령어
\$ mkdir scpdd	테스트 디렉토리
\$ touch scptest	테스트 파일
\$ scp scptest ubuntu@localhost:/home/ubuntu/scpdd	???
\$ scp -P 22 scptest ubuntu@localhost:/home/ubuntu/scpdd	???
\$ scp -r scpdd ubuntu@localhost:/home/ubuntu/clon	???

네트워크 설정 방법

▪ 네트워크 관리 필수 명령어 – netstat (option)

- a : 현재 다른 PC와 연결(Established)되어 있거나 대기(Listening) 중인 모든 포트 번호 확인
- r : 라우팅 테이블 확인 및 커넥션되어 있는 포트번호를 확인
- n : 현재 다른 PC와 연결되어 있는 포트번호를 확인(IP주소로 화면 출력)
- e : 랜카드에서 송수신한 패킷의 용량 및 종류를 확인
- t : tcp protocol
- u : udp protocol
- l : Listening 중인 포트 번호 확인
- p : 해당 프로토콜을 사용하는 프로그램, 프로세스 ID를 보여줌
- c : 1초 단위로 보여줌

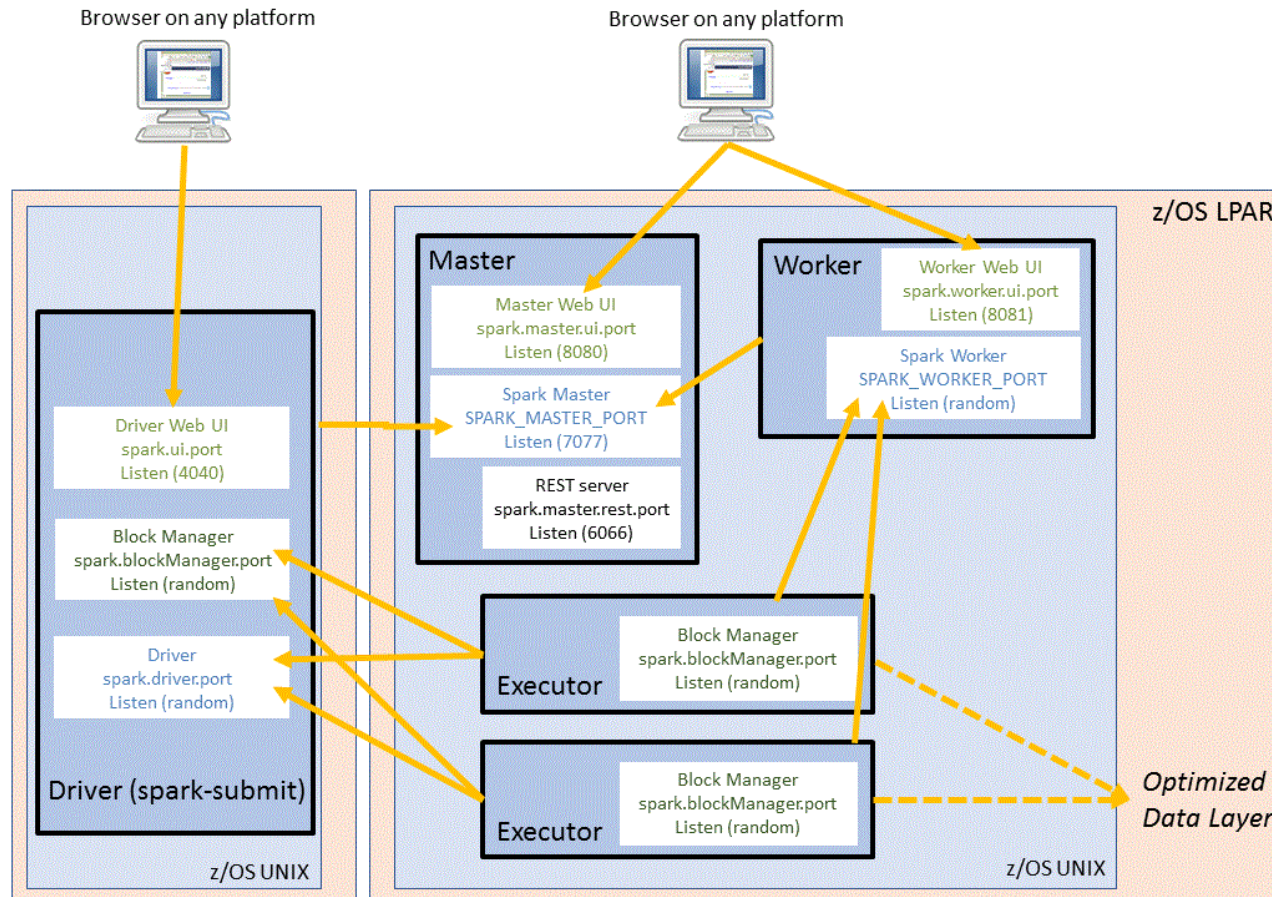
```
$ sudo netstat -nltp
```

현재 다른 PC와 연결되어 있는 Listen 상태의 tcp protocol을 PID와 함께 출력

```
ubuntu@ubuntu:/etc/ssh$ sudo netstat -nltp
[sudo] password for ubuntu:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*                LISTEN      795/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      1210/sshd
tcp6       0      0 :::80                   :::*                    LISTEN      3688/apache2
tcp6       0      0 :::22                   :::*                    LISTEN      1210/sshd
ubuntu@ubuntu:/etc/ssh$
```


방화벽 설정

■ 일반적인 Apache Spark Cluster 환경에서의 Service Port 사용



Port Name	Port Number
Master Web UI	8080
Worker Web UI	8081
History Server Web UI	18080
Master port	7077
Master REST port	6066
Worker port	(random)
Block manager port	(ramdom)
Shuffle server	7337

※ source : <https://www.ibm.com>

방화벽 설정

▪ ufw (ubuntu 기본 방화벽)

- ubuntu 의 기본 방화벽이며 iptables를 쉽게 설정할 수 있도록 구현됨.
- 간단한 방화벽 구성은 가능하나 수준 높은 방화벽 구성에는 iptables 룰을 직접 사용해야 함

▪ ufw 활성화/비활성화

\$ sudo ufw enable	ufw 활성화
\$ sudo ufw disable	ufw 비활성화
\$ sudo ufw status	ufw 상태 확인
\$ sudo ufw status numbered	ufw 상태 정보에 번호 추가

▪ ufw 기본 룰 설정

\$ sudo ufw show raw	기본 룰 내용 확인
\$ sudo ufw default allow	나가는 패킷 전부 허가(allow)
\$ sudo ufw default deny	들어오는 패킷 전부 거부(deny)

방화벽 설정

▪ ufw 허용 및 차단

\$ sudo ufw allow 22	ssh 서비스 포트 22번 허용(tcp/udp)
\$ sudo ufw deny 22	ssh 서비스 포트 22번 차단(tcp/udp)
\$ sudo ufw allow 22/tcp	tcp 22번 포트만 허용
\$ sudo ufw allow 22/udp	udp 22번 포트만 허용
\$ sudo ufw allow ssh	서비스 명으로도 설정 가능
\$ sudo ufw allow from 192.168.0.1 to any port 22	
\$ sudo ufw allow from 192.168.0.1 to any port 3306	

▪ ufw 룰의 삭제/리셋/리로드

\$ sudo ufw delete deny 22	해당 룰의 내용을 삭제
\$ sudo ufw reset	ufw 기본값으로 복원
\$ sudo ufw reload	ufw 룰 리로드

6. Device 관리 명령어

- 1) mount 사용법
- 2) disk 관리 명령어

mount 사용법

▪ mount 명령어

- 이동식 미디어 장치(CD-ROM, USB 메모리 등)를 액세스 하기 위해 특정한 위치에 연결해 주는 과정을 마운트 라고함
- 기본적으로 mount 명령은 현재 시스템에 마운트 된 미디어 장치의 목록을 표시

▪ Mount 사용법

```
# mount -t <FileSystem type> <Device File> <Mount point>
```

```
# umount <Device File, Mount Point>
```

```
$ sudo mount -t iso9660 /dev/cdrom /media/cdrom
```

시디롬 마운트

```
$ sudo umount /dev/cdrom /media/cdrom
```

시디롬 마운트 해제

```
$ sudo mount -o loop ~/test.iso /media/iso
```

하나의 iso파일 마운트

```
$ sudo umount ~/test.iso /media/iso
```

iso파일 마운트 해제

▪ File System Type

vfat	윈도우의 긴 파일시스템
ntfs	윈도우 NT, XP 및 Vista, 7, 10에서 사용되는 윈도우 고급 파일시스템
iso9660	표준 CD-ROM 파일 시스템

disk 관리 명령어

Linux 디스크 관리

- df – 각각의 장치에 얼마나 많은 디스크 공간이 있는지 알고 싶을 때 df명령으로 마운트된 모든 디스크의 상황을 쉽게 볼 수 있음.
- du – 디스크 공간이 부족할 때 그 상황을 쉽게 볼 수 있음. 특정 디렉토리(기본값은 현재 디렉토리)의 디스크 사용량을 보여줌

\$ df	디스크 사용 현황을 출력 (1024 byte block 수로 공간을 보여줌)
\$ df -h	human-readable 형태로 출력(K/M/G)
\$ du	현재 위치 디렉토리 디스크 사용현황
\$ du -c	나열된 모든 파일의 총계를 출력
\$ du -h	human-readable 형태로 출력(K/M/G)
\$ du -s	각 인수를 요약하여 출력

7. Environment Variable 설정

1) 환경 변수 설정

환경 변수 설정

Linux 환경변수

- 프로그램과 스크립트에서 시스템 정보를 얻고 임시 데이터 및 구성정보를 저장하기 위해 환경변수를 사용하여 Linux Shell 환경을 정의하는데 도움

환경변수	구분	설명
/etc/environment	시스템	시스템 단계에서 설정하는 환경변수 파일이며 모든 사용자 적용
/etc/profile	시스템	시스템 bash셸의 주요 기본 시동 파일이며 모든 사용자 적용
/etc/profile.d/*.sh	시스템	시스템 구동시 실행될 모든 환경변수 파일을 저장함
\$HOME/.bashrc	사용자	사용자의 특정 환경변수를 정의하기 위해서 사용(대화형 셸)
\$HOME/.profile	사용자	사용자의 특정 환경변수를 정의하기 위해서 사용(로그인 셸)

기본 환경변수 수정 예

```
$ printenv
$ vi .profile
JAVA_HOME=/usr/local/javahome
export JAVA_HOME
PATH=$PATH:/home/envtest
```

현재 설정된 모든 환경변수 보기
 사용자 환경변수인 .profile 파일 열기
 맨 아랫줄에 JAVA_HOME 추가
 환경변수 적용
 맨 아랫줄에 PATH 내용 추가

```
$ source .profile
```

변경된 내용을 적용

환경 변수 설정

■ 기본 셸 환경변수

변수	설명	
CDPATH	cd명령에 대한 검색 경로로 사용되는 콜론으로 구분된 디렉토리 목록	
HOME	현재 사용자의 홈 디렉토리	
IFS	셸이 텍스트 문자열을 분할할 때 사용하는 필드를 구분하는 문자들의 목록	
MAIL	현재 사용자의 메일박스 파일 이름(bash 셸에서 새 메일이 왔는지 확인하기 위해 검사)	
MAILPATH	현재 사용자의 받은 메일함에 대한 콜론으로 구분된 여러 파일 이름의 목록 (bash 셸에서 새 메일이 왔는지 확인하기 위해 검사)	
PATH	셸이 명령을 찾을 때 쓸 콜론으로 구분된 디렉토리 목록	
PS1	기본 셸 커맨드라인 인터페이스 프롬프트 문자열 \$ sudo adduser ubuntu3 <terminal 2로 이동> ubuntu3로 로그인 \$ export PS1='Wu@Wh:WwW\$ '	Wu 사용자 이름 Wh 호스트 이름 WH 전체 호스트 이름 W! 현재 명령의 history 번호 W\$ root이면 #, 일반 사용자는 \$ Ww 현재 작업 디렉토리 WW 현재 작업 디렉토리 전체 경로 Wt 현재 시간을 HH:MM:SS로 출력 Wd 현재 날짜를 Mon Jun 20로 출력 Ws Shell 이름
PS2	보조 셸 커맨드라인 인터페이스 프롬프트 문자열	

8. APT Package 관리 도구

- 1) APT(Advanced Package Tool) 명령어
- 2) Repository 설정방법

APT(Advanced Package Tool) 명령어

▪ APT 명령어

- 사용자 편의성을 위해 Ubuntu 패키지 관리 툴인 apt-get과 apt-cache를 통합한 명령어
- 완전한 통합이 아니므로 일반사용자는 apt 사용이 효과적이지만 세밀한 옵션을 사용하기 위해서는 apt-get를 사용해야함

▪ apt 명령과 apt-get/apt-cache 명령어

apt	apt-get	설명
apt install	apt-get install	패키지 목록
apt remove	apt-get remove	패키지 삭제(관련 파일은 유지됨)
apt purge	apt-get purge	패키지 삭제(관련 파일까지 제거)
apt update	apt-get update	리파지토리 인덱스 갱신
apt upgrade	apt-get upgrade	모든 패키지 업그레이드
apt autoremove	apt-get autoremove	불필요한 패키지 제거(미사용 패키지)
apt full-upgrade	apt-get dist-upgrade	의존성 패키지 통합 업그레이드
apt search	apt-cache search	프로그램 검색
apt show	apt-cache show	패키지 상세 정보 출력
apt list	apt-get install	apt 명령어로 apt-get install 과 동일(wc, grep 활용)
apt edit-sources	/etc/apt/sources.list	apt 명령어로 소스 리스트를 편집할 때 사용.

Repository 설정

▪ apt 패키지 관리 도구 Repository

- /var/lib/apt/lists – Repository 패키지 목록을 업데이트
- /etc/apt/sources.list – Repository 목록을 관리하기 위한 용도로 사용
- /etc/apt/source.list.d/ - Repository 목록을 파일형태로 관리하기 위한 용도로 사용

```
## Note, this file is written by cloud-init on first boot of an instance
## modifications made here will not survive a re-bundle.
## if you wish to make changes you can:
## a.) add 'apt_preserve_sources_list: true' to /etc/cloud/cloud.cfg
##     or do the same in user-data
## b.) add sources in /etc/apt/sources.list.d
## c.) make changes to template file /etc/cloud/templates/sources.list.tpl

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://archive.ubuntu.com/ubuntu bionic main restricted
deb-src http://archive.ubuntu.com/ubuntu bionic main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://archive.ubuntu.com/ubuntu bionic-updates main restricted
deb-src http://archive.ubuntu.com/ubuntu bionic-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://archive.ubuntu.com/ubuntu bionic universe
deb-src http://archive.ubuntu.com/ubuntu bionic universe
deb http://archive.ubuntu.com/ubuntu bionic-updates universe
deb-src http://archive.ubuntu.com/ubuntu bionic-updates universe
```

9. Process 및 Resource 관리 명령어

- 1) Process 관리
- 2) Service 관리
- 3) System Resource 관리 방법

Process 관리

▪ ps 명령어

- 현재 시스템에서 실행되고 있는 process를 보여주는 가장 기본적인 명령어
- USER ID, CPU 사용량, MEMORY 사용량, 사용한 명령어 등의 정보를 보여줌

\$ ps -e	모든 실행중인 process 목록
\$ ps -fu ubuntu	지정한 사용자 full-format process 목록
\$ ps -ef	
\$ ps -e -o pid,uname,pcpu,pmem	지정된 항목의 내용만 출력
\$ ps -e -sort=-pcpu head -5	cpu 사용량으로 정렬하여 5개 프로세스 목록 출력

▪ 프로세스 강제 종료

# ubserver 계정	
\$ ping www.google.com &	백그라운드 프로그램 실행
# ubuntu 계정	
\$ ps -ef grep ubserver	ubserver 실행 프로세스 검색
\$ sudo kill [PID]	프로세스 강제 종료

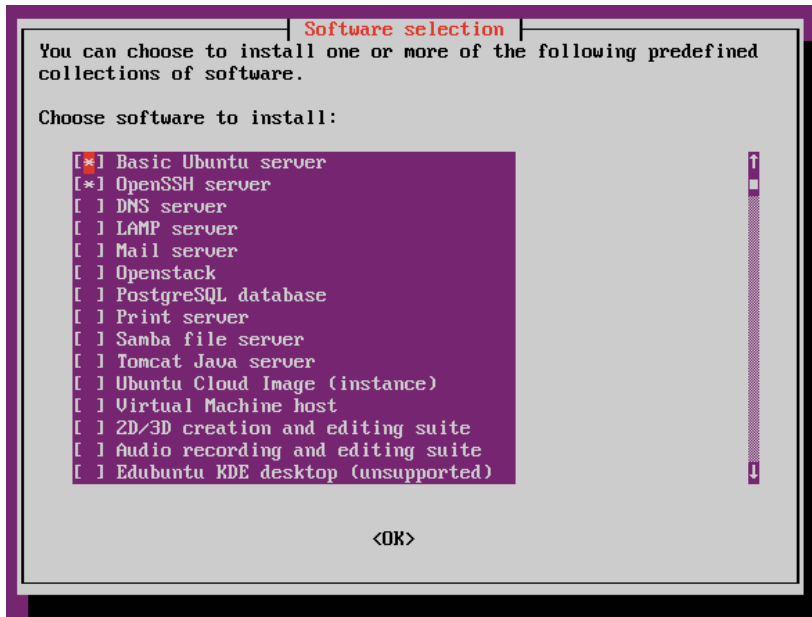
Service 관리

▪ service 명령어

- /etc/init.d 에 있는 service init script 및 service 데몬을 실행/중지/재실행 하는 유틸리티
- 부팅시 실행되는 네트워크 설정, iptables 방화벽 설정, sshd, httpd 등의 서비스 프로그램 설정

```
$ service --status-all
$ service start sshd
$ service stop sshd
$ service restart sshd
```

모드 서비스의 상태를 출력
sshd 서비스 시작
sshd 서비스 중지
sshd 서비스 재시작



```
acpid          cryptdisks-early  lvm2
apache2        dbus              lvm2-lvmetad
apache-htcacheclean ebttables         lvm2-lvmpolld
apparmor       grub-common       lxcfs
appport        hwclock.sh        lxd
atd            irqbalance        mdadm
console-setup.sh iscsid             mdadm-waitidle
cron           keyboard-setup.sh networking
cryptdisks     kmod              open-iscsi
```

```
open-vm-tools  udev
plymouth       ufw
plymouth-log   unattended-upgrades
procps         uuidd
rsync          x11-common
rsyslog        xinetd
screen-cleanup
ssh
sysstat
```

System Resource 관리 방법

주요 System Resource 관리 도구

```
top - 15:23:39 up 7:57, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 95 total, 1 running, 57 sleeping, 0 stopped, 0 zombie
kCPU(s): 0.0 us, 0.3 sy, 0.0 ni, 99.3 id, 0.3 wa, 0.0 hi, 0.0 si, 0.0 st
kIB Mem + 1009124 total, 442132 free, 100876 used, 455116 buff/cache
kIB Swap: 2017276 total, 2017276 free, 0 used, 760404 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR   S  %CPU  %MEM     TIME+ COMMAND
 4024 Ubuntu    20   0   45328    588    519   S  0.0   0.0   0:00.04 sshd
4113 ubserver   20   0   40372   3792   3232   R  0.3   0.4   0:00.01 top
    1 root      20   0   77640   6628   0   S  0.0   0.0   0:01.58 systemd
    2 root      20   0      0      0      0   S  0.0   0.0   0:00.00 kthreadd
    4 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 kworker/0:0H
    6 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 mm_percpu_wq
    7 root      20   0      0      0      0   S  0.0   0.0   0:00.17 ksoftirqd/0
    8 root      20   0      0      0      0   I  0.0   0.0   0:00.48 rcu_sched
    9 root      20   0      0      0      0   I  0.0   0.0   0:00.00 rcu_bh
   10 root      rt  0      0      0      0   S  0.0   0.0   0:00.00 migration/0
   11 root      rt  0      0      0      0   S  0.0   0.0   0:00.15 watchdog/0
   12 root      20   0      0      0      0   S  0.0   0.0   0:00.00 cpuhp/0
   13 root      20   0      0      0      0   S  0.0   0.0   0:00.00 kdevtmpfs
   14 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 netns
   15 root      20   0      0      0      0   S  0.0   0.0   0:00.00 rcu_tasks_kthre
   16 root      20   0      0      0      0   S  0.0   0.0   0:00.00 kauditd
   17 root      20   0      0      0      0   S  0.0   0.0   0:00.01 khungtaskd
   18 root      20   0      0      0      0   S  0.0   0.0   0:00.00 oom_reaper
   19 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 writeback
   20 root      20   0      0      0      0   S  0.0   0.0   0:00.00 kcompactd0
   21 root      25   5      0      0      0   S  0.0   0.0   0:00.00 ksm
   22 root      39 19   0      0      0   S  0.0   0.0   0:00.00 khugepaged
   23 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 crypto
   24 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 kintegrityd
   25 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 kblockd
   26 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 ata_sff
   27 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 md
   28 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 edac-poller
   29 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 devfreq_wq
   30 root      0 -20   0      0      0   I  0.0   0.0   0:00.00 watchdogd
```

top - 시스템 모니터링 도구

```
total DISK READ : 0.00 B/s | total DISK WRITE : 168.70 K/s
actual DISK READ: 0.00 B/s | actual DISK WRITE: 0.00 B/s
  PID  PRIO  USER      DISK READ  DISK WRITE  SWAPIN      IO    COMMAND
3342 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.01 % [kworker/0:0]
382 be/3 root      0.00 B/s    161.03 K/s  0.00 %    0.00 % systemd-journald
1126 be/4 syslog    0.00 B/s    7.67 K/s    0.00 %    0.00 % rsyslogd -n [rs:main Q:Reg]
    1 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % init maybe-ubiquity
    2 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kthreadd]
    4 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kworker/0:0H]
    6 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [mm_percpu_wq]
    7 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [ksoftirqd/0]
    8 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [rcu_sched]
    9 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [rcu_bh]
   10 rt/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [migration/0]
   11 rt/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [watchdog/0]
   12 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [cpuhp/0]
   13 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kdevtmpfs]
   14 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [netns]
   15 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [rcu_tasks_kthre]
   16 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kauditd]
   17 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [khungtaskd]
   18 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [oom_reaper]
   19 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [writeback]
   20 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kcompactd0]
   21 be/5 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [ksmd]
   22 be/7 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [khugepaged]
   23 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [crypto]
   24 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kintegrityd]
   25 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kblockd]
   26 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [ata_sff]
   27 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [md]
   28 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [edac-poller]
   29 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [devfreq_wq]
   30 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [watchdogd]
   34 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [ksuadd0]
   35 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [ecryptfs-kthrea]
   77 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kthrotid]
```

iotop - 입출력 모니터링 도구

```
CPU: 0.7% Tasks: 37, 75 thr: 1 running
Mem: 99.8M/985M Load average: 0.00 0.00 0.00
Swap: 0K/1.92G Uptime: 07:57:53

  PID USER      PRI  NI  VIRT   RES   SHR   S  %CPU  %MEM     TIME+ Command
4114 ubserver   20   0 25552  4272  5552   R  0.7   0.4   0:00.04 httpd
4091 ubserver   20   0 105M  3400  2436   R  0.3   0.3   0:00.03 sshd: ubserver@pts/0
    1 root      20   0 77640  8772  6628   S  0.0   0.3   0:01.58 /sbin/init maybe-ubiquity
382 root      19  -1 124M 25252 24432   S  0.0   2.5   0:00.40 /lib/systemd/systemd-journald
416 root      20   0 46676  5604  3168   S  0.0   0.6   0:00.89 /lib/systemd/systemd-udevd
418 root      20   0 31708  1744  1572   S  0.0   0.2   0:00.00 /sbin/iptables -f
828 systemd-n 20   0 80012  5380  4788   S  0.0   0.5   0:00.06 /lib/systemd/systemd-networkd
860 systemd-r 20   0 70740  4984  4428   S  0.0   0.5   0:00.07 /lib/systemd/systemd-resolved
1124 syslog    20   0 261M  4516  3720   S  0.0   0.4   0:00.01 /usr/sbin/rsyslogd -n
1125 syslog    20   0 261M  4516  3720   S  0.0   0.4   0:00.00 /usr/sbin/rsyslogd -n
1126 syslog    20   0 261M  4516  3720   S  0.0   0.4   0:00.02 /usr/sbin/rsyslogd -n
975 syslog    20   0 261M  4516  3720   S  0.0   0.4   0:00.05 /usr/sbin/rsyslogd -n
989 root      20   0 76640  6216  5292   S  0.0   0.6   0:00.16 /lib/systemd/systemd-logind
991 messagebu 20   0 50164  4836  4004   S  0.0   0.5   0:00.39 /usr/bin/dbus-daemon --system --addr
1308 root      20   0 165M 17132  9296   S  0.0   1.7   0:00.00 /usr/bin/python3 /usr/bin/networkd-d
1023 root      20   0 165M 17132  9296   S  0.0   1.7   0:00.10 /usr/bin/python3 /usr/bin/networkd-d
1028 root      20   0 157M 1676 1544   S  0.0   0.2   0:00.00 /usr/bin/iptables -t
1027 daemon    20   0 28332  2348  2140   S  0.0   0.2   0:00.00 /usr/sbin/iptables -f
1050 root      20   0 157M 1676 1544   S  0.0   0.2   0:00.00 /usr/bin/iptables -var/lib/iptables/
1051 root      20   0 157M 1676 1544   S  0.0   0.2   0:00.00 /usr/bin/iptables -var/lib/iptables/
1028 root      20   0 157M 1676 1544   S  0.0   0.2   0:00.00 /usr/bin/iptables -var/lib/iptables/
1460 root      20   0 470M 26604 16420   S  0.0   2.6   0:00.15 /usr/lib/snapd/snapd
1461 root      20   0 470M 26604 16420   S  0.0   2.6   0:00.00 /usr/lib/snapd/snapd
1462 root      20   0 470M 26604 16420   S  0.0   2.6   0:00.00 /usr/lib/snapd/snapd
1474 root      20   0 470M 26604 16420   S  0.0   2.6   0:00.00 /usr/lib/snapd/snapd
1477 root      20   0 470M 26604 16420   S  0.0   2.6   0:01.00 /usr/lib/snapd/snapd
1482 root      20   0 470M 26604 16420   S  0.0   2.6   0:01.20 /usr/lib/snapd/snapd
1072 root      20   0 470M 26604 16420   S  0.0   2.6   0:02.71 /usr/lib/snapd/snapd
1138 root      20   0 273M 7096 6116   S  0.0   0.7   0:00.64 /usr/lib/accounts-service/accounts-da
1158 root      20   0 273M 7096 6116   S  0.0   0.7   0:00.01 /usr/lib/accounts-service/accounts-da
```

htop - 강화된 top 도구

```
12.5kb 25.0kb 37.5kb 50.0kb 62.5kb
Ubuntu <= hkg12s01-ln-f100.1e100.net 672b 672b 857b
                                     672b 672b 840b

  PID  PRIO  USER      DISK READ  DISK WRITE  SWAPIN      IO    COMMAND
 3342 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.01 % [kworker/0:0]
 382 be/3 root      0.00 B/s    161.03 K/s  0.00 %    0.00 % systemd-journald
1126 be/4 syslog    0.00 B/s    7.67 K/s    0.00 %    0.00 % rsyslogd -n [rs:main Q:Reg]
    1 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % init maybe-ubiquity
    2 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kthreadd]
    4 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kworker/0:0H]
    6 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [mm_percpu_wq]
    7 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [ksoftirqd/0]
    8 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [rcu_sched]
    9 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [rcu_bh]
   10 rt/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [migration/0]
   11 rt/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [watchdog/0]
   12 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [cpuhp/0]
   13 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kdevtmpfs]
   14 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [netns]
   15 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [rcu_tasks_kthre]
   16 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kauditd]
   17 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [khungtaskd]
   18 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [oom_reaper]
   19 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [writeback]
   20 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kcompactd0]
   21 be/5 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [ksmd]
   22 be/7 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [khugepaged]
   23 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [crypto]
   24 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kintegrityd]
   25 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kblockd]
   26 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [ata_sff]
   27 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [md]
   28 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [edac-poller]
   29 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [devfreq_wq]
   30 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [watchdogd]
   34 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [ksuadd0]
   35 be/4 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [ecryptfs-kthrea]
   77 be/0 root      0.00 B/s    0.00 B/s    0.00 %    0.00 % [kthrotid]

  cum: 7.76KB peak: 1.31KB rates: 672b 672b 857b
  avg: 7.82KB    1.64KB      672b 672b 840b
TOTAL: 15.6KB 2.95KB 1.31KB 1.31KB 1.66KB
```

iftop - 네트워크 패킷 모니터링 도구

10. Remote Access 및 SSH 키 생성을 통한 인증

- 1) SSH key 개요
- 2) SSH key 생성 및 설정
- 3) SSH를 이용한 접속

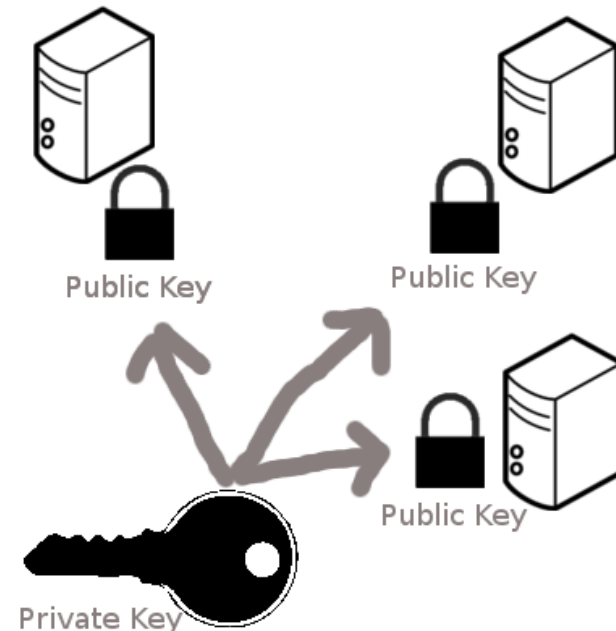
SSH key 개요

SSH Key

- 서버에 원격으로 접속 할 때 비밀번호 대신 key를 통해 접속하는 방식
- 비밀번호보다 높은 수준의 보안이 필요하거나 로그인 없이 자동 접속 시 사용
- SSH key는 공개키(public key)와 비공개 키(private key)로 두 개의 키를 생성함

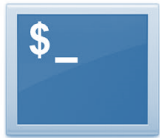


1. `ssh-keygen` 으로 key pair 생성
 - `id_rsa` (private key)
 - `id_rsa.pub` (public key)
2. server로 public key 복사
3. 인증파일로 PK 내용 추가
 - `authorization_keys`
4. client에서 server로 ssh 접속



SSH key 생성 및 설정

SSH key 생성 및 설정 방법



ubuntu

```
$ sudo adduser ubserver  
  
$ ssh-keygen -t rsa  
$ ls -al ~/.ssh/  
$ cd .ssh  
  
$ scp id_rsa.pub ubserver@localhost:~/
```

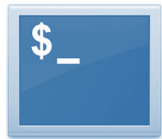


ubserver

```
$ ls -al  
$ mkdir .ssh  
$ chmod 700 .ssh  
$ cd .ssh  
  
$ touch authorized_keys  
$ chmod 644 authorized_keys  
  
$ cat ~/id_rsa.pub >> authorized_keys
```

SSH를 이용한 접속

SSH를 이용한 접속



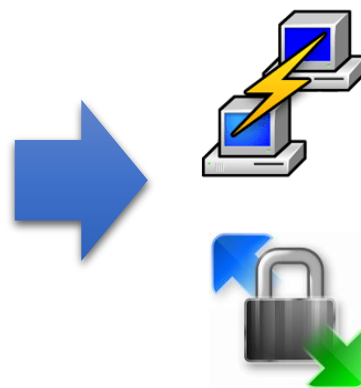
ubuntu

```
$ whoami  
$ ssh ubserver@localhost  
  
$ whoami  
$ w
```

Host에서 putty를 이용한 Guest 접속방법

HOST
(Windows or OS X)

1. NAT 및 Host Only Adapter 설정
2. Port Forwarding 설정
- 22 port



원격접속

파일전송

GUEST
(ubuntu)



11. Log 관리 파일 및 명령어

- 1) Log 관리 File의 종류
- 2) Log 관리 명령어

Log 관리 File의 종류

Log file 의 종류

- 기본적인 로그들은 syslogd에 의해 제어가 되며, syslogd의 설정파일인 /etc/syslog.conf 파일을 수정함으로써 이 파일들의 저장위치와 저장 파일명 변경이 가능

로그이름	로그 파일명	관련 데몬	설명
커널 로그	/dev/console	kernel	콘솔에 뿌려지는 로그
시스템 로그	/var/log/messages	syslogd	리눅스 커널로그 및 주 로그
보안 로그	/var/log/secure	xinetd	보안 인증 관련 로그
메일 로그	/var/log/maillog	sendmail popper	메일로그(sendmail)
크론 로그	/var/log/cron	crond	crond에 의한 로그
부팅 로그	/var/log/boot.log	kernel	시스템 부팅 로그
커널 부트메시지 로그	/var/dmesg	kernel	시스템 부팅 메시지 로그
커널 로그	/var/log/wtmp	kernel	시스템 전체 로그인 기록
커널 로그	/var/log/utmp	kernel	현재 로그인 사용자 기록
FTP 로그	/var/log/xferlog	ftpd	ftp 로그
웹 로그	/var/log/httpd/access_log	httpd	아파치(웹서버) 로그
웹 로그	/var/log/httpd/error_log	httpd	아파치(웹서버) 에러
네임서버 로그	/var/log/named.log	named	네임서버(DNS) 로그

Log 관리 명령어

▪ 콘솔 로그 (/dev/console)

- 커널에 관련된 내용을 시스템 콘솔에 뿌려주는 로그이다.
- 시스템에 관련된 중요한 내용들(시스템폴, 다운 등)에 대한 로그를 관리자에게 알리고자 함이 목적이다.
- 출력을 파일로 저장하는 것이 아니라 장치명(/dev/console)을 사용하여 콘솔로 로그를 뿌려준다.
- timestamp, 호스트명, 커널 메시지 내용 등이 기록 되었다.

▪ 시스템 로그 (/var/log/messages)

- 사용명령어 : dmesg (/var/log/messages를 출력한다.)
- 주로 접속 시 인증에 관한 것과 메일에 관한 내용, 시스템에 관한 변경사항 등 시스템에 관한 전반적인 로그를 기록하는 파일이다.
- timestamp, 호스트명, 데몬명, 메시지 내용 등이 기록된다.
- 시스템 관리자에 의해서 가장 소중하게 다루어지는 로그이다.
- 보안사고가 발생시에 가장 먼저 분석을 해야하는 파일이다.
- 메시지 내용은 su 실패 로그, 특정 데몬 비활성화 로그, 부팅 시 발생 에러 등 다양한 로그 포함
- syslog facility에 의해 남은 로그로 /etc/syslog.conf 설정에 따라 남는 정보의 종류가 달라짐

Log 관리 명령어

▪ 보안 로그 (/var/log/secure)

- 모든 접속과 관련하여 언제 어디서 어떤 서비스를 사용했는지 기록한다.
- timestamp, 호스트명, 응용프로그램명(pid), 메시지 내용이 기록되어있다.
- 보통 login, tcp_wrappers, xinetd 관련 로그들이 남는다.
- ps -ef라는 옵션 외에도 aux라는 옵션으로 확인 가능하다. (예 : ps aux | grep xinetd)
- 실행중인 xinetd의 PID저장 파일은 /var/run/xinetd.pid

▪ 메일 로그 (/var/log/maillog)

- sendmail이나 pop등의 실행에 관한 기록이다.
- 메일을 주고받을 때에 이 로그파일(smtp, pop)에 기록이다
- 실행중인 sendmail의 PID저장 파일은 /var/run/sendmail.pid
- timestamp, 호스트명, 데몬명(pid), 메시지 내용 기록

Log 관리 명령어

▪ 크론 로그 (/var/log/cron)

- 시스템의 정기적인 작업(cron)에 대한 모든 작업한 기록을 보관하고 있는 파일이다.
- 크론데몬의 crond가 언제 어떤작업을 했는가를 확인 가능하다.
- crond의 의해서 실행되었던 데몬(프로세스, 응용프로그램 등)들이 기록 되었다.
- 실행중인 crond의 PID저장 파일은 /var/run/crond.pid
- /etc/ 디렉토리 밑에 있는 cron.hourly, crondaily, cron.weekly, cron.monthly 파일들에 기록되어 있는 작업을 실행한 후에 cron 파일에 log를 기록한다.
- timestamp, 호스트명, 데몬명(pid), 메시지 내용이 기록되어 있다.

▪ 부팅로그 (/var/log/boot.log)

- 시스템의 데몬들이 실행되거나 재시작되었을 때 기록되는 로그 파일이다.
- 부팅 시의 에러나 조치 사항을 확인할 때 활용이 가능하다.
- timestamp, 호스트명, 데몬명(pid), 메시지 내용이 기록 된다.

Log 관리 명령어

▪ 커널 부트 메시지 로그 (/var/dmesg)

- 시스템이 부팅할 때 출력되었던 메시지를 로그 기록한다.

▪ /var/log/wtmp

- 사용 명령어 : last
- 사용자들의 로그인-로그아웃 정보 기록이다
- 바이너리 형태이며 지금까지 사용자들의 로그인, 로그아웃 히스토리를 모두 누적형태로 저장된다.
- 시스템의 셧다운, 부팅 히스토리까지 포함한다.
- 해킹 피해 시스템 분석 시 비중있게 다룬다.

옵션	설명
\$ last [계정명]	계정명을 입력하면 사용자별 로그 정보를 출력
\$ last -f [파일명]	지난 파일에 대해서 로그를 점검시 -f옵션 뒤에 해당 파일명을 입력
\$ last -R	IP를 제외시킨 로그 정보 출력
\$ last -a	로그 정보를 출력할 때 IP를 뒤로 배치하여 출력
\$ last -d	외부에서 접속한 정보와 reboot 정보만을 출력

Log 관리 명령어

▪ /var/log/utmp

- 사용자 명령어 : who, w, whodo, uesrs, finger
- 시스템에 현재 로그인한 사용자들에 대한 상태를 기록한다.
- 리눅스에서는 /var/run 혹은 /var/adm, 솔라리스에서는 /etc등에 위치하며 바이너리 형태로 저장되어 vi 편집기 등으로 읽을 수 없다.
- utmp(x) 파일은 기본적으로 사용자 이름, 터미널 장치 이름, 원격 로그인 시 원격 호스트 이름, 사용자 로그인한 시간 등을 기록 한다.
- w 명령어는 utmp(x)를 참조하여 현재 시스템에 성공적으로 로그인한 사용자에게 대한 snapshot을 제공해 주는 명령어이다.
- 해킹 피해 시스템 분석 시 비중있게 다룬다.

※ wtmp, wtmpx와 파일 포맷은 동일 ump(x)는 현재 시스템에 대한 정보, wtmp(x)는 누적된 정보

▪ /var/log/lastlog

- /etc/passwd 파일에 정의되어 있는 모든 계정의 최근 접속 정보를 확인 가능하다.
- 사용자의 최근 로그인 시간을 사용자 이름, 터미널, IP 주소, 마지막 로그인 시간 출력
- /var/log/lastlog 파일에 저장되고 바이너리 형태 (-u(접속이름), -t(접속시간), -h(도움말))

Log 관리 명령어

▪ FTP 로그 (/var/log/xferlog)

- ftp나 ncftp 등의 접속이 이루어 졌을 때 이 로그파일에 기록이 된다.
- ftp를 사용했을 때 이 로그파일에 기록되고, 업로드 파일과 다운로드한 파일들에 대한 자세한 정보가 기록 저장된다.

▪ 웹 로그 (/var/log/httpd/access_log, /var/log/httpd/error_log)

▶ Access log

- 웹사이트에 접속했던 사람들이 각 파일들을 요청했던 실적을 기록해놓은 목록을 저장한다.
- 방문자의 IP또는 도메인 네임, 방문자가 파일을 요청한 시간, 방문자가 웹서버에 요청한 처리 내용(Get, Put, Head), 방문자가 요구한 파일의 이름, 파일의 크기 및 처리결과 등의 데이터를 제공

▶ Error log

- 요청한 홈페이지가 없거나 링크가 잘못되는 등의 오류가 있을 경우에 생성된다.

▪ /var/log/btmp

- 사용자 명령어 : lastb
- 로그인 시도 5번 이상 실패한 로그 기록을 확인 가능하다.
- 계정명, 접속 콘솔/터미널 유무, IP, 시간 정보 출력
- /var/log/btmp에 바이너리 형태로 저장도니다.

Log 관리 명령어

▪ History (해당 계정의 home directory/ .bash_history)

- 접속한 계정에서 사용했던 명령어의 내용만 보여준다.
- root의 경우 ~/.bash_history에 사용한 명령어가 저장된다.
- 저장되는 로그의 위치를 변경하려면 export HISTFILE="경로/파일이름" 을 입력 한다.

▪ Pacct (/var/account/pacct)

- 사용자 명령어 : lastb
- 시스템에 들어온 사용자가 어떤 명령어를 실행시키고 어떠한 작업을 했는지에 대한 사용 내역 등이 기록 된다.
- 사용된 명령어의 argument와 그 명령어가 시스템 내 어느 파일 시스템의 어느 디렉토리에 실행 되었는지는 기록되지 않는다.
- /var/account/pacct에 바이너리 파일로 기록된다.
- 파일 크기가 쉽게 커지기 때문에 관리가 필요한 파일 이다.

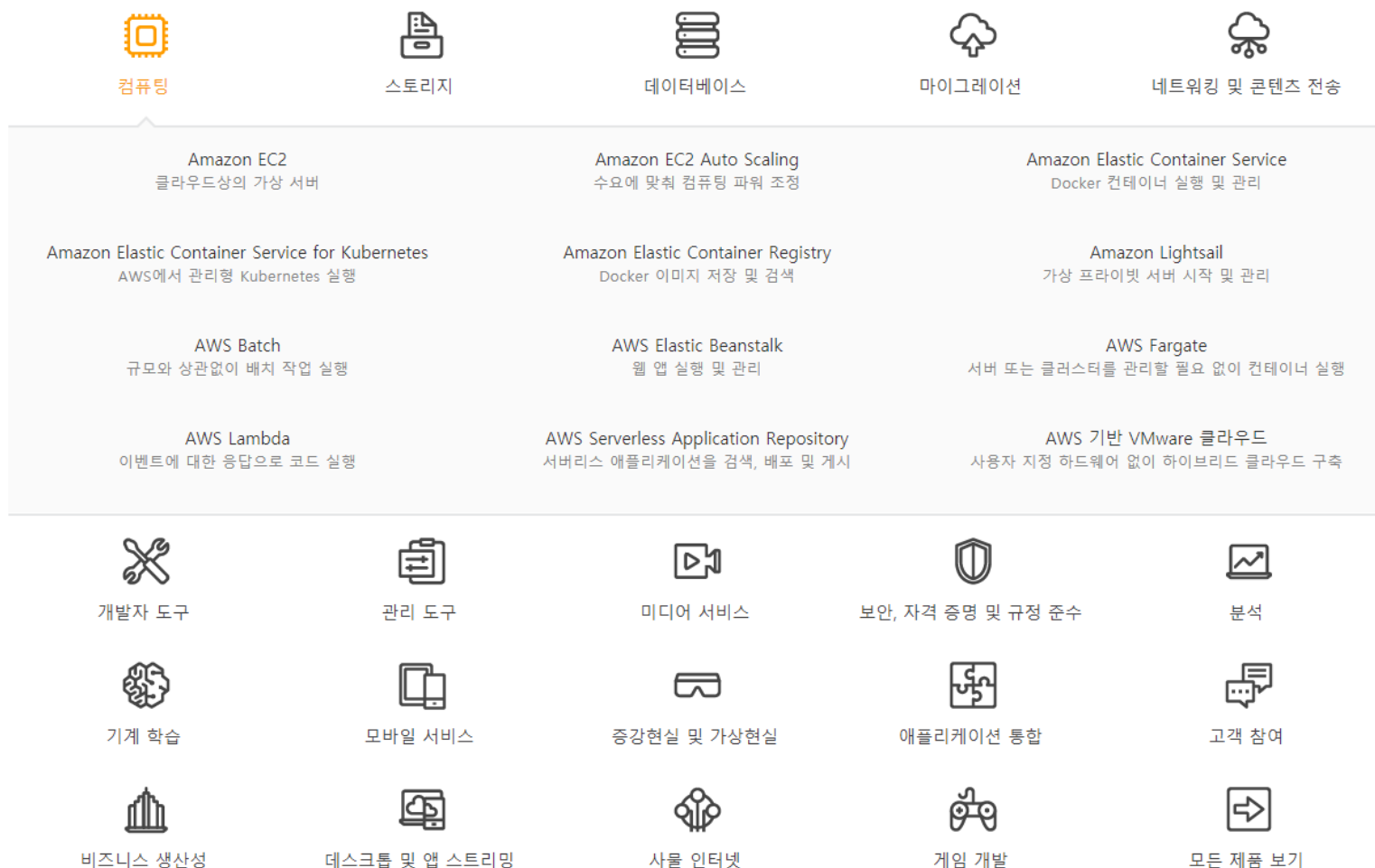
12. Amazon Web Service 이용 방법

- 1) AWS 소개
- 2) AWS Management Console
- 3) AWS Instance(VM) 생성 및 원격 접속 방법

AWS 소개

Amazon Web Services는 IAAS 형태로 제공되는 Public cloud service

- <http://aws.amazon.com>



AWS Management Console

■ AWS 서비스 사용을 위한 통합 관리 콘솔

The screenshot displays the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, a dropdown menu for '서비스' (Services), a dropdown for '리소스 그룹' (Resource Groups), a star icon, a notification bell, and user information 'Hwangno Lee' with a dropdown for '서울' (Seoul) and a '지원' (Support) link.

The main content area is divided into several sections:

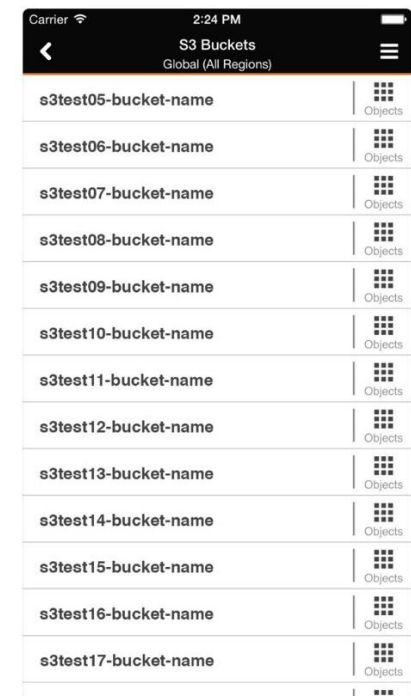
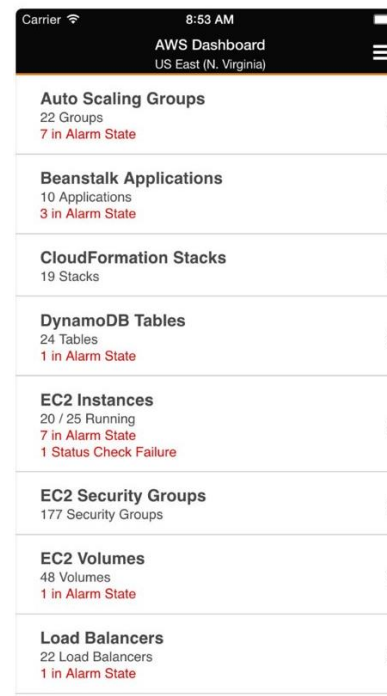
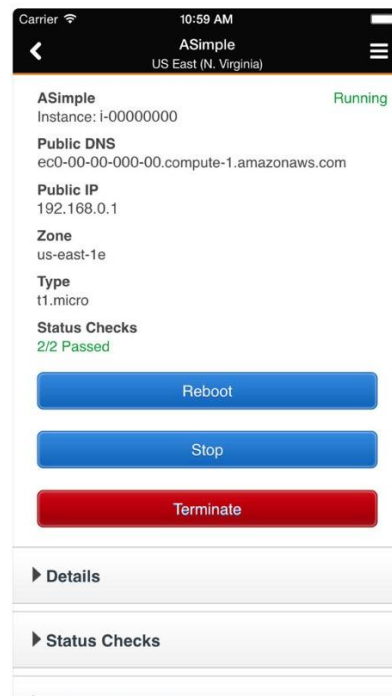
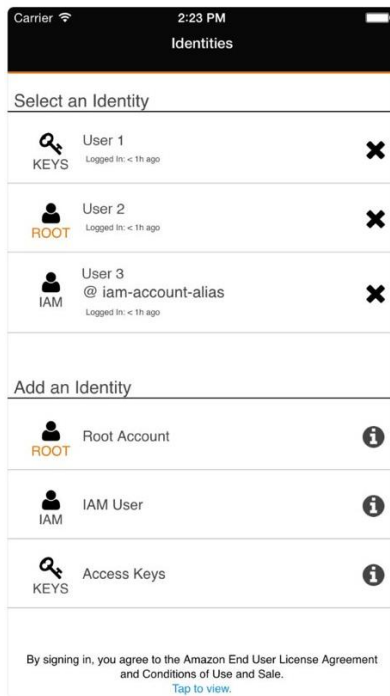
- AWS 서비스 (AWS Services):** A search bar with the placeholder text '이름 또는 기능(예: EC2, S3 또는 VM, 스토리지)으로 서비스를 찾습니다.' (Find services by name or function (e.g., EC2, S3 or VM, storage)). Below the search bar, there's a section for '최근 방문한 서비스' (Recently visited services) with icons for '결제' (Billing), 'IAM', '비용 탐색기' (Cost Explorer), and 'AWS Organizations'. A link for '모든 서비스' (All services) is also present.
- 솔루션 구축 (Solution Architecture):** A section titled '간단한 마법사와 자동화된 워크플로우로 시작합니다.' (Get started with simple wizards and automated workflows). It features six tiles:
 - 가상 머신 시작** (Start Virtual Machine): Using EC2, ~2-3 minutes.
 - 웹 앱 구축** (Build Web App): Using Elastic Beanstalk, ~6 minutes.
 - 가상 서버를 이용하여 구축** (Build using Virtual Server): Using Lightsail, ~1-2 minutes.
 - IoT 디바이스 연결** (Connect IoT Device): Using AWS IoT, ~5 minutes.
 - 개발 프로젝트 시작** (Start Development Project): Using CodeStar, ~5 minutes.
 - 도메인 등록** (Register Domain): Using Route 53, ~3 minutes.
- 유용한 팁 (Useful Tips):**
 - 비용 관리** (Cost Management): AWS Budgets를 사용하여 AWS 비용, 사용량 및 예약을 모니터링합니다. [지금 시작](#) (Get started now).
 - 조직 생성** (Create Organization): 여러 AWS 계정을 정책 기반으로 관리하기 위해 AWS Organizations을 사용합니다. [지금 시작](#) (Get started now).
- AWS 탐색 (AWS Explorer):**
 - Amazon SageMaker를 사용한 기계 학습** (Machine Learning using Amazon SageMaker): 기계 학습 모델을 구축, 교육 및 배포하는 가장 빠른 방법. [자세히 알아보기](#) (Learn more).
 - Amazon Relational Database Service (RDS)**: RDS는 사용자를 대신해 데이터베이스를 관리하고 확장/축소하며 Aurora, MySQL, PostgreSQL.

At the bottom of the '솔루션 구축' section, there is a '더 보기' (View more) link.

AWS Management Console

▪ Mobile App을 통한 AWS Console 지원

- Google Playstore, Apple Appstore에서 다운로드 가능
- dashboard를 지원하여 중요 알림이나 생성된 Instance에 대한 제어를 할 수 있음



AWS Instance(VM) 생성 및 원격 접속 방법

인스턴스 생성 및 원격 접속

- Putty 프로그램을 이용하여 AWS 내 생성된 Instance에 접속 가능함.

<div> Launch Instance Connect Actions ▾ </div>							
<div> <input type="text"/> Filter by tags and attributes or search by keyword ? < </div>							
<input type="checkbox"/>	Name ▾	Instance ID ▲	Instance Type ▾	Availability Zone ▾	Instance State ▾	Status Checks ▾	Alarm Status
<input checked="" type="checkbox"/>	CodeCommit...	i-0c7116dc	m3.large	us-east-1b	● running	✓ 2/2 checks ...	✓ OK
<input type="checkbox"/>	WinStack - a...	i-17aba838	c3.large	us-east-1a	● stopped		None
<input type="checkbox"/>	RoadTripBlog...	i-7053641e	m1.small	us-east-1b	● running	✓ 2/2 checks ...	None
<input type="checkbox"/>	PCoIP Conn...	i-7714d98d	m3.medium	us-east-1b	● stopped		None
<input type="checkbox"/>	Dyna-Dyna-P...	i-92115b41	t1.micro	us-east-1b	● running	✓ 2/2 checks ...	None
<input type="checkbox"/>	WinStack - a...	i-98a5a6b7	c3.large	us-east-1a	● stopped		None
<input type="checkbox"/>	ECS Instanc...	i-afb00351	t1.micro	us-east-1c	● running	✓ 2/2 checks ...	None

Thank You

