Ansible 자동화로 구현한 클라우드 환경 DevOps

- 그룹 보고서 5번 문제 풀이 -

제출일	2020. 12. 31
훈련기관	㈜솔데스크
강사님	김기연 강사님
조원	1조 최시울, 김민영, 황유지, 허윤하, 김선경

목 차

개 요

이 보고서는 K-Digital 수업 LINUX I, II 파트를 마치며 그룹별로 주어진 문제에 대해 고민해 보고 답을 작성해 정리한 보고서입니다.

저희 1조가 맡은 질문은 “파일 관리 및 디렉터리 관리 시 원격지 관리와 로컬 관리의 차이점과 시스템에서 특정한 파일을 검색하는 경우 자주 사용될 옵션과 그 예시”입니다.

원격접속과 로컬접속의 정의를 다루고 파일과 디렉토리를 관리할 때 어떠한 차이점이 존재하는지 기재했으며 원격접속과 로컬접속 시 환경 파일의 특징에 대해 자세하게 다뤘습니다. 또한, grep, find 명령어와 함께 자주 사용하는 옵션과 예시, alias 사용 예시를 들어 업무를 어떻게 더 효율적으로 할 수 있는지에 초점을 맞추었습니다.

거시적으로 이 모든 내용을 다루어 어떤 업무에 적용할 수 있을지 고민한 결과, 환경 파일(/etc/profile)에 alias를 적어서 선언시켜준 뒤, 로컬접속 사용자나 원격접속 사용자 모두 효율적으로 업무를 할 수 있도록 하는 결과를 도출했습니다. 추가로 실무에서 사용자들이 자주 찾는 단어나 파일을 grep, find 명령어를 이용해 마찬가지로 같은 과정을 거쳐 유용하게 쓸 수 있습니다.

소프트웨어 환경 및 버전

소프트웨어	설명
LINUX	CentOS 8.3
VM	VMware Workstation 15.5.6

보고서 포맷

전체 보고서 : 맑은 고딕 10pt, 줄 간격 150%
쉘 작성 : 굴림체 8pt, 표 안에 작성

원 격 접 속 과 로 컬 접 속

원격접속과 로컬접속 정의

"굳이 원격접속을 사용하지? 그냥 노트북 들고 다니면서 로컬접속하는 것이 낫지 않을까?"사실 현업에 있거나 전공자가 아니면 생각할 수 있는 의문점입니다. **원격접속이란 무엇일까? 바로 다른 서버에 네트워크를 연결하여 로그인하는 것입니다. 쉽게 말하면 내 친구 컴퓨터에 서버에 내 컴퓨터로 접속을 하는 것입니다.** 윈도우 운영체제에서도 기본으로 설치되어 있는 원격 제어 프로그램을 실행 할 수 있고, 요즘 자주 사용하는 '팀뷰어'라는 프로그램도 원격 접속 프로그램입니다.

실제로 대부분의 회사는 서버실이 따로 있습니다. 보안상 거의 들어가지 못하기 때문에 이러한 이유로 원격 접속으로 관리를 합니다. 또한 출장 근무자, 재택근무자 등 많은 이유가 있으며, 컴퓨터의 성능 차이도 있기도 합니다. 공간의 제약이 없어 효율적으로 서버를 관리할 수가 있기 때문에 원격접속은 어쩌면 로컬접속보다 더 자주 사용됩니다.

로컬접속은 간단하게 물리적으로 나의 집, 사무실에 있는 컴퓨터, 노트북으로 직접 서버에 접속하는 것입니다.

본질적으로 원격접속 또한 시간과 비용을 아끼는 효율적인 프로그램입니다. 극단적인 예시지만, 해외에 있는 서버 접속을 위해 해외에 출장을 수시로 나간다거나 회사 본사로 출근해서 서버를 사용하는 것보단 집에서 원격접속으로 재택근무를 하는 것이 많은 시간과 비용을 아낀다는 것을 알 수 있습니다.

원격접속의 종류와 변화

예전부터 rlogin, telnet, rsh와 파일전송을 위한 rcp, ftp를 사용해 왔습니다. 하지만, 원격접속을 할 때, 평문으로 전달되기 때문에 누군가가 접근해서 비밀번호, 개인 정보 등을 훔쳐갈 수 있는 위험성이 컸습니다. 특히나, rsh 같은 경우는 누구든지 접속이 가능해서 보안에 취약한 단점이 있습니다. 이러한 문제를 보완한 것이 ssh 명령어입니다.

ssh 명령어는 Secure shell로 서버와 호스트 사이에 인증키를 받는 동시에, 암호화 기능을 추가시켜 보안을 강화했습니다. ssh(openssh-clients 패키지)에는 scp, sftp 프로그램이 있어 원격하고자 하는 서버로 파일을 전송하거나 받을 수 있으며, rcp, ftp보다 상대적으로 안전합니다.

원격접속 명령어와 예시

ssh: 'secure shell', 1995년 핀란드인 타투 일료넨은 'secure shell communication'라는 회사를 설립해 'secure shell'을 개발하고 상용화했습니다. 네트워크 상에서 두 호스트 사이의 **통신을 암호화해서 안전한 접속과 통신을 제공하는 프로토콜**입니다. 원격접속에서는 가장 중요한 부분이 인증 방법입니다. 패스워드를 사용하여 인증하는데, 이 인증 과정도 **모두 암호화되어 패스워드가 노출이 되지 않습니다.**

scp: 'secure copy', ssh를 설치하면 같이 설치되며, 따로 ftp 클라이언트를 설치하지 않아도 파일전송을 할 수 있습니다. **데이터를 암호화해서 전송**하기 때문에 scp 프로그램의 단점을 보완합니다. 단일 파일을 가져오거나 내보내고, 이름을 지정해서 많이 사용합니다.

sftp: 'secure File Transfer Protocol', ftp보다 안전하며 인터넷으로 연결하여 파일을 전송하거나 다운로드합니다. 파일을 전송할 때는 파일 정보를 마찬가지로 **암호화합니다. ftp는 상대적으로 대규모 파일을 가져올 때 사용합니다.**

1. ssh 명령어 형식

```
# ssh user01@192.168.10.252
# ssh 192.168.10.252 ls -l file1
# ssh 192.168.10.252 server2.example.com
# ssh 192.168.10.252 ls /test
```

2. ssh 명령어로 다른 서버의 사용자로 접속

```
#ssh fedora@192.168.10.252
 The authenticity of host '192.168.10.252 (192.168.10.252)' can not be established.
ECDSA key fingerprint is SHA256:VupU+fz6cK1zkTDfQxGuWW4Ylt+aTy5yXNvxsVtHwz4.
ECDSA key fingerprint is MD5:27:8e:59:e4:ac:31:a7:4d:c6:92:88:bc:74:5c:45:c7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.10.252' (ECDSA) to the list of known hosts.
fedora@192.168.10.252's password:
Last login: Tue Mar 24 12:01:34 2020 from ::ffff:192.168.10.200

#hostname
server2.example.com
#id
uid=1001(fedora)                    gid=1001(fedora)                    groups=1001(fedora)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

3. scp 명령어

```
1. server1에서 파일을 server2로 전달하기

# systemctl status sshd -> sshd가 실행 중이 아니면 사용할 수 없기 때문에 확인해봅니다.

# cd /test
# cd /etc/group grouplist.txt
# echo "checked on 201230" >> grouplist.txt
# cat grouplist.txt

# scp /test/grouplist.txt root@192.168.10.252:/tmp
The authenticity of host '192.168.10.252 (192.168.10.252)' can not be established.
ECDSA key fingerprint is SHA256:VupU+fz6cK1zkTDfQxGuWW4Ylt+aTy5yXNvxsVtHwz4.
ECDSA key fingerprint is MD5:27:8e:59:e4:ac:31:a7:4d:c6:92:88:bc:74:5c:45:c7.
```

```
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.10.252' (ECDSA) to the list of known hosts.
root@192.168.10.252 is password:
grouplist.txt 100% 2274 1.8MB/s 00:00

# ssh 192.168.10.252 ls /tmp/*txt*
root@192.168.10.252's password:
/tmp/grouplist.txt
/tmp/linux200.txt

2. server2에서 파일을 server1로 받아오기

# ssh user01@192.168.10.252 ls /test
user01@192.168.10.252 is password:
file1
-------> 연결 끊고, 다시 server1 계정으로 복귀

# scp user01@192.168.10.252:/home/user01/test/file1 /test
user01@192.168.10.252 is password:
file1 100% 890 823.1KB/s 00:0
# ls /test
file1

이 뿐만 아니라 scp -r 명령어로 디렉토리와 내부 파일까지 모두 가져오거나 보낼 수 있습니다.
```

4. sftp 명령어 형식

```
# sftp [계정]@[상대방주소]:[소스경로]


server 1

* root계정

# cd /test
# pwd
/test


server 2

$ id
uid=1001(fedora)                    gid=1001(fedora)                    groups=1001(fedora)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
$ cd /test
$ cp /etc/services file1
$ cp file1 file2
$ cp file1 file3

server 1
```

```
# sftp fedora@192.168.10.252:/test/*                    ---------> 받는 경로를 설정 안함.
fedora@192.168.10.252 is password:
Connected to 192.168.10.252.
Fetching /test/file1 to file1
/test/file1 100% 655KB 57.7MB/s 00:00
Fetching /test/file2 to file2
/test/file2 100% 655KB 62.5MB/s 00:00
Fetching /test/file3 to files3
/test/file3 100% 655KB 43.7MB/s 00:00

# ls /test
file1    file2     file3     backupfile     happynewyear2021
```

반대로 server2의 fedora에게 파일을 보낼 때

```
# stfp fedora@192.168.10.252                  ----------> 소스 경로가 없으면 입력 서버로 원격접속
fedora@192.168.10.252's password:
Connected to 192.168.10.252                  --------> 이미 접속한 기록이 있어 경고창이 재출력 안됨
sftp> help                                   ------>  sftp> 명령행에서 사용 가능한 명령어 목록 출력
Available commands:
bye Quit sftp
cd path Change remote directory to 'path'
chgrp [-h] grp path Change group of file 'path' to 'grp'
chmod [-h] mode path Change permissions of file 'path' to 'mode'
chown [-h] own path Change owner of file 'path' to 'own'
df [-hi] [path] Display statistics for current directory or
 filesystem containing 'path'
exit Quit sftp
get [-afPpRr] remote [local] Download file
reget [-fPpRr] remote [local] Resume download file
reput [-fPpRr] [local] remote Resume upload file
help Display this help text
lcd path Change local directory to 'path'
lls [ls-options [path]] Display local directory listing
lmkdir path Create local directory
ln [-s] oldpath newpath Link remote file (-s for symlink)
lpwd Print local working directory
ls [-1afhlnrSt] [path] Display remote directory listing
lumask umask Set local umask to 'umask'
mkdir path Create remote directory
progress Toggle display of progress meter
put [-afPpRr] local [remote] Upload file
pwd Display remote working directory
quit Quit sftp
rename oldpath newpath Rename remote file
rm path Delete remote file
rmdir path Remove remote directory
symlink oldpath newpath Symlink remote file
```

```
version Show SFTP version
!command Execute 'command' in local shell
! Escape to local shell
? Synonym for help


sftp> pwd                                                    ----------> 원격 접속한 서버 위치
Remote working directory: /home/fedora
sftp>  cd /test
sftp>  pwd
/test
sftp> ls
file1     file2      file3     Kdigital.txt
      (server1 root 계정으로 보냈던 file* 파일들을 볼 수 있습니다.)
sftp> put happynewyear2021
sftp> ls
file1     file2     file3     Kdigital.txt     happynewyear2021
                                                -----> server1 root 계정의 파일이 복사된 것

주소와 소스 경로말고도 sftp 명령어로 파일을 다운받을 수도 있습니다.
다운 받을 때는 mget을 입력하면 됩니다.

# sftp user01@192.168.10.252
user01@192.168.10.252's password:
Connected to 192.168.10.252
sftp> pwd
Remote working directory: /home/user01
sftp> cd ~/dir1
sftp> pwd
/home/user01/dir1
sftp> ls
ByeBye2020      HappyNewYear
sftp> mget /home/user01/ByeBye2020                           ----------> mget 명령어와 경로 입력
Fetching /home/user01/ByeBye2020
sftp> lpwd
Local working directory: /test
sftp> quit

# ls
file1     file2      file3      backupfile      happynewyear2021      ByeBye2020          ---> 다운 완료
```

파일, 디렉토리 관리 시 원격지 관리와 로컬 관리의 차이점

파일과 디렉토리 관리 시 원격지에서 관리하는 것과 로컬 관리의 차이점은 없다고 봐야합니다. 앞에서 다뤘던 많은 원격 명령어의 예시를 보면 권한이 걸린 디렉토리나 파일도 똑같은 권한을 부여받으며, 환경 파일의 설정들도 똑같이 제어를 받습니다. 사실 이 문제를 어렵게 생각하여 차이점을 찾으려고 고민을 할 수 있습니다. 하지만, 팀뷰어를 통해 다른 컴퓨터를 연결해서 메모장을 작성하고, 새로운 폴더를 생성한다고 생각하면 이해가 명확하게 될 것입니다.

환 경 파 일

환경 파일이란?

환경 파일은 사용자가 로그인하거나 새로운 쉘을 실행할 때 자동으로 실행되는 명령을 저장한 파일입니다. 시스템 환경 파일과 사용자 환경 파일이 있으며 쉘마다 다른 이름의 파일을 사용합니다. 시스템 환경 파일은 전체 사용자의 공통 환경을 설정하는 파일이고, 사용자 환경 파일은 각 사용자의 홈 디렉터리에 숨김 파일로 생성되어 사용자가 내용을 수정하고 관리할 수 있습니다.

환경 파일의 종류와 역할

bash는 다섯 개의 설정 파일이 존재하며 각 파일의 역할은 아래와 같습니다.

/etc/profile : 환경 변수와 bash가 수행될 때 실행되는 프로그램을 제어하는 전역적인 시스템 설정과 관련된 파일입니다.

```
# vi /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

...중략...
unset i
unset -f pathmunge

if [ -n "${BASH_VERSION-}" ] ; then
 if [ -f /etc/bashrc ] ; then
 # Bash login shells run only /etc/profile
 # Bash non-login shells run only /etc/bashrc
 # Check for double sourcing is done in /etc/bashrc.
 . /etc/bashrc
 fi
fi
```

/etc/bashrc : 별칭(alias)과 bash가 수행될 때 실행되는 함수를 제어하는 전역적인 시스템 설정과 관련된 파일입니다. 때로는 생략되어 /etc/profile에 포함되기도 합니다.

```
# vi /etc/bashrc
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile
.
```

.bash_profile : 환경 변수와 bash가 수행될 때 실행되는 프로그램을 제어하는 지역적인 시스템 설정과 관련된 파일입니다. 이들 환경 변수는 오직 그 사용자에게만 한정되며, 그 이외의 다른 사람에게는 영향을 미치지 않습니다. 이 파일은 전역적인 설정 파일인 /etc/profile이 수행된 다음 바로 수행됩니다.

```
# vi .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
  . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
```

.bashrc : 별칭(alias)과 bash가 수행될 때 실행되는 함수를 제어하는 지역적인 시스템 설정과 관련된 파일입니다. 이들 별칭과 함수들은 오직 그 사용자에게만 한정되며, 그 이외의 다른 사람에게는 영향을 미치지 않습니다. 이 파일은 전역적인 설정 파일인 /etc/bashrc이 수행된 다음 바로 수행됩니다.

```
# vi .bashrc
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
alias cdt='cd /test'
alias rmt='cd /test && rm -rf *'

# Source global definitions
if [ -f /etc/bashrc ]; then
  . /etc/bashrc
fi
```

.bash_logout : 사용자가 로그아웃하기 바로 직전에 실행하는 프로그램에 관한 bash의 지역적인 시스템 설정과 관련된 파일입니다. 이들 프로그램은 오직 그 프로그램을 실행하는 사용자에게만 영향을 끼치지 다른 사람에게는 아무런 영향을 주지 않습니다.

로컬 접속 시 환경 정책

1. umask 027로 조정

일반 사용자들이 파일을 생성할 때 다른 사용자가 해당 파일이나 디렉터리를 마음대로 수정하거나 읽을 수 없도록 처음부터 일반 사용자에 대한 권한을 주지 않도록 합니다. 계정의 Start Profile(/etc/profile)에 명령을 추가하면, 사용자가 새로운 쉘에서 로그인 한 후에도 변경된 umask 값을 적용받게 됩니다.

```
# vi /etc/profile
# /etc/profile


...중략...
xport PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL


# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
 umask 002
else
 umask 027                                                    ----------> 수정
fi
...중략...
fi
```

2. 일반 사용자 su 명령어 사용 제한

su 명령어는 기본적으로 사용자 전환에 사용되며, su –로 접근 시 해당 계정의 권한까지 사용합니다. 따라서 root 계정으로 접근할 수 있는 su 명령어를 일반 사용자들이 사용할 수 없도록 하는 설정이 필요합니다.

```
# ls -l /usr/bin/su
-rwsr-xr-x. 1 root root 50320 Jul 22 01:55 /usr/bin/su
# chmod 4750 /usr/bin/su ; ls -l /usr/bin/su
-rwsr-x---. 1 root root 50320 Jul 22 01:55 /usr/bin/su
```

원격 접속 시 환경 정책

1. SSH / telnet 설치 유무 확인

```
# rpm -qa | grep open-ssh

# rpm -qa | grep -c telnet
```

참고) 원격 서버에 ssh 프로토콜을 사용하여 터미널에 접속하려면 아래와 같은 조건이 필요합니다.

- **22번 tcp 포트가 방화벽에서 열려있어야 한다.**
- **SSH 서버 프로그램이 설치 및 구동 되고 있어야 한다.**
- **SSH 프로토콜로 접속할 수 있는 클라이언트가 필요하다.**

2. 원격 접속 시 로그인 관련 환경 파일

▼ 로그인 시 읽히는 환경 파일 순서

```
1.[1] /etc/profile

2. ~/.bash_profile, ~/.bash_login , ~/.profile

3. ~/.bashrc

4. [2] /etc/bashrc
```

▼ 원격접속 시 로그인 쉘 출력

```
[root@server1 ~]# ssh root@localhost

root@localhost's password:
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Thu Dec 10 18:37:17 2020 from 192.168.10.1

|---> /etc/profile read

|---> [3] /etc/profile.d/*.sh test.sh read

|---> /etc/bashrc read

|---> /etc/bashrc out

|---> ~/.bash_profile read
```

1) 보통 /etc/profile 에서는 /etc/bash.bashrc를 읽어 들인다.
2) /etc/bashrc에는 모든 계정이 사용하는 설정들이 들어가 있다.

```
|---> ~/.bashrc read

|---> /etc/bashrc read

|---> /etc/bashrc out

|---> ~/.bashrc out

|---> ~/.bash_profile out
```

3 로그 아웃 시 출력되는 환경 파일

```
# exit

|---> ~/.bash_logout read
```

3) /etc/profile.d 디렉토리 안에 있는 모든 쉘 스크립트(*sh)를 실행시키는 명령어 코드이다.

파 일 검 색

grep 명령어

grep은 하나 이상의 패턴과 일치하는 행을 선택하여 주어진 파일 및 디렉터리를 탐색합니다. 해당 명령어의 옵션에는 여러 가지가 있으며 옵션으로 사용해도 되고 egrep, fgrep과 같이 다른 명령어 표현으로 사용해도 됩니다. 또한, 해당 명령어는 |(파이프)를 통해 사용할 수도 있습니다.

[옵션]
-l : 패턴이 있는 파일 이름만 출력
-n : 패턴을 포함하는 줄을 줄번호와 함께 출력
-v : 패턴을 포함하는 줄을 제외하고 출력
-c : 패턴을 찾은 줄의 수 출력
-i : 패턴을 찾을 때 대소문자 구분 안 함

```
# grep [옵션] [패턴] [파일] 또는 CMD | grep [옵션] [패턴]
```

grep을 통해 검색하는 방법 중 메타 문자를 사용하면 편리합니다. 맨 앞에서 해당 패턴으로 시작하는 경우는 ^기호를 사용하고 맨 뒤에서 해당 패턴으로 끝나는 경우는 $기호를 사용합니다.

```
# grep root /etc/passwd ( # cat /etc/passwd | grep root )
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
# grep ^root /etc/passwd ( # cat /etc/passwd | grep ^root )
root:x:0:0:root:/root:/bin/bash
# grep bash$ /etc/passwd ( # cat /etc/passwd | grep bash$ )
root:x:0:0:root:/root:/bin/bash
team01:x:1000:1000:team01:/home/team01:/bin/bash
.
.
```

두 가지 이상의 패턴을 검색할 경우 egrep 명령어(또는 grep의 –E 옵션)를 이용합니다. 특수기호까지 검색을 원한다면 grep 명령어(또는 grep의 –f 옵션)를 이용합니다.

```
# grep -E 'tema01|root' /etc/passwd ( egrep 'team01|root' /etc/passwd )
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
team01:x:1000:1000:team01:/home/team01:/bin/bash
```

```
# cat file2
^hello
hello
hello^
# fgrep '^hello' file2                    ----------> ^기호까지 검색 가능
^hello
# grep '^hello' file2                     ----------> ^기호 검색 불가능
hello
hello^
```

find 명령어

find는 실시간으로 검색하여 파일의 위치를 찾아주는 명령어입니다. 해당 명령어에는 다양한 옵션들을 사용할 수 있습니다. 파일 사용 권한, 파일의 유형, 크기, 수정시간, 이름, anode 번호 등의 기준으로 검색할 수 있습니다. 또한, 원하는 조건을 두 개 이상 표현이 가능합니다. 아래의 예제를 통해서 확인할 수 있습니다.

[옵션]
-name : 파일 이름
-iname : 대소문자를 구별하지 않고 파일 이름
-perm : 파일 권한
-type : 파일의 종류
-size : 파일의 크기
-links : 링크 개수
-user : 사용자 ID
-group : 그룹 ID
-atime : 특정 기간 접근하지 않은 파일
-mtime : 특정 기간 수정하지 않은 파일
-inum : 지정된 inode 번호
-exec : 찾은 파일에 다른 명령어 실행

```
# find [검색 위치] [옵션1] [인자1] [옵션2] [인자2] …
```

```
# find / -name “file”
최상위 디렉터리에서 file이라는 이름의 파일검색
# find / -type l -iname “*FILE*”
최상위 디렉터리에서 file이라는 이름(대소문자 구분 없음)의 링크 검색
# find / -perm -400
최상위 디렉터리에서 읽기 이상의 권한을 가진 파일검색
# find / -user team01
최상위 디렉터리에서 소유자가 team01인 파일검색
# find . mtime -2 -type f
현재 디렉터리에서 수정날짜가 이틀이 안 된 파일검색
```

예시 1) 오래된 로그 기록의 경우 주기적으로 삭제

```
# find / -name “*.log” -mtime +30 -exec rm -r {} W;
 최상위 디렉터리에서 이름이 .log로 끝나는 파일 중 수정일이 30일이 넘은 파일을 삭제
```

예시 2) 현재 디렉터리 안에 있는 파일 중 setuid를 가지는 파일의 권한을 755로 변경

```
# ls -l
total 0
-rwsr-xr-x. 1 root root 0 Dec 30 16:12 file1
# find . -perm -4000 -exec chmod 755 {} W;
# ls -l
total 0
-rwxr-xr-x. 1 root root 0 Dec 30 16:12 file1
```

예시 3) 여러 개의 압축파일 풀기

```
# find . -name “*.tar.gz” -exec tar zxcf {} W;
tar.gz확장자로 끝나는 파일들을 압축해지
```

업무 효율을 위한 alias 설정

원하는 파일을 찾기 위해서 find라는 명령어와 grep이라는 명령어를 사용했습니다. 그러나 매번 같은 작업을 해야 하는 경우 로그 파일을 자주 열어봐야 하거나, 또는 명령어가 너무 길어 매번 치기 번거로운 경우 alias 명령어를 사용하여 쉬고 간편하게 실행할 수 있습니다. alias를 입력하여 현재 설정되어 있는 명령어들을 확인할 수 있습니다.

```
# alias
alias cp='cp -i'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias mv='mv -i'
alias rm='rm -i'
alias rmt='cd /test && rm -rf *'
alias which='(alias; declare -f) | /usr/bin/which --tty-only --read-alias --read-functions
--show-tilde --show-dot'
alias xzegrep='xzegrep --color=auto'
alias xzfgrep='xzfgrep --color=auto'
alias xzgrep='xzgrep --color=auto'
alias zegrep='zegrep --color=auto'
alias zfgrep='zfgrep --color=auto'
alias zgrep='zgrep -color=auto'
```

아래의 예시는 logfind라는 명령어의 입력 시 최상위 디렉터리에서 이름이 .log로 끝나는 파일을 검색하는 alias를 뜻합니다. 필요에 따라 원하는 명령어로 교체해 주면 됩니다.

```
# alias logfind=' find / -name *.log
```

alias 영구 설정

보통은 위 명령어 형태로 쓰이며, 단순히 alias 명령어를 쉘에서 실행할 경우 해당 사용자만 사용할 수 있습니다. 쉘을 빠져나가거나, 다른 사용자가 사용하기엔 적절하지 않습니다. 해당 alias를 환경 파일에 설정을 함으로써 영구적으로 다른 사용자가 사용할 수 있게 구성할 수 있습니다. 다음의 두 가지 방법 중 적절한 방법을 선택합니다.

첫 번째, ~/.bashrc 파일에 설정합니다. 본인의 홈 디렉터리 안의 .bashrc파일을 이용하는 것입니다. 이 경우 쉘을 빠져나갔다가 들어와도 alias를 사용할 수 있습니다. vi 편집기를 이용하여 .bashrc파일에 지정하고 싶은 alias 명령어를 입력합니다.

```
# cd
# vi .bashrc
# .bashrc

# User specific aliases and functions
```

```
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
alias logfind=' find / -name *.log'                    ----------> 설정하려는 alias 입력

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
~
```

두 번째, /etc/bashrc 파일에 설정합니다. 이 경우 전체 사용자가 쓸 수 있고 쉘을 빠져나갔다가 들어와도 alias를 사용할 수 있습니다.

```
# vi /etc/bashrc
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile
.
.
alias logfind=' find / -name *.log'                    ----------> 설정하려는 alias 입력
```

예시) 오래된 로그 파일을 정기적으로 확인 후 삭제해 주는 명령어를 alias로 생성하고 해당 alias를 매번 생성하지 않고 모든 사용자가 사용할 수 있도록 /etc/bashrc 파일에 설정합니다.

```
# cd
# pwd
/root
# vi /etc/bashrc
# .bashrc
# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
alias cdt='cd /test ; pwd'
alias rmt='cd /test && rm -rf *'
alias logcheck=' find / -name  "*.log" -mtime +60 -exec rm -r {} W;'        ---> 설정하려는 alias 입력

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
~
```

결 론

마무리

로컬접속과 원격접속을 할 때 환경파일들이 열리면서 많은 설정과 정책이 확인되고 실행된다. 이에 따라 특정 계정이 환경 파일을 설정해 자신의 기호대로 사용할 수 있으며, 모든 사용자에게 자주 쓰이는 명령어들은 특정 환경파일에 넣어 서버를 접속하는 모든 사용자들이 서버를 편리하고 효율적으로 이용할 수 있다.

참조 문서

- 리눅스 부팅과정 참고 문서 :
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/performing_a_standard_rhel_installation/getting-started_installing-rhel
https://www.linux.co.kr/lecture/lec_linux_01/lec-data/08data.pdf

- SWAP 참고 문서 :
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/managing_storage_devices/getting-started-with-swap_managing-storage-devices#swap-space_getting-started-with-swap

- 원격접속 사용 이유 :
https://limetimeline.tistory.com/37

- 원격접속과 파일전송 수업 내용 :
https://www.notion.so/b12c7592d7c54029bba09aad5df4191e#2628c72be7b241148c7d12a265d03440

- ssh 명령어의 유래와 정의 :
https://terms.naver.com/entry.nhn?docId=3351634&cid=40942&categoryId=32828

- umask 참조 문서 :
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_basic_system_settings/file-permissions-rhel8_configuring-basic-system-settings

- 환경 파일 참조 문서
https://www.ibm.com/support/knowledgecenter/ssw_aix_72/filesreference/environment.html

- 원격접속 환경 파일 참조 문서
https://originalchoi.tistory.com/23