리눅스 수업

Linux – file & directory

명령어

-> ls : 현재 파일의 목록을 보여줌

-> ls -l : 현재 디렉토리 파일과 디렉토리를 좀더 자세히 보여주는 명령어

-> pwd : 현재 위치 표시

-> touch [파일명] : 빈 파일 생성

서버 구축

1. Virtualbox에 리눅스 설치

2. 장치 – 게스트 확장 CD 이미지 삽입 선택후 설치

3. 장치 – 클립보드 공유 및 드래그 앤 드롭 양방향 설정

**Vsftpd 설치**

1. sudo apt-get update

2. sudo apt-get install vsftpd

3. /etc/vsftpd.conf 파일 설정

4. sudo gedit /etc/vsftpd.conf

아래 내용으로 변경

#inetd 모드 NO standalone 모드 YES

listen=YES

#inetd 모드 NO standalone 모드 YES ip6일때

listen\_ipv6=NO

#익명 접속 허용 여부 설정

anonymous\_enable=NO

#데이터 전송을 위해 Active Mode 사용 여부 설정

port\_enable=NO

#데이터 전송을 위해서 Passive mode를 사용할 것인지 설정

pasv\_enable=YES

#로컬 계정 사용자들의 접속을 허용할 것인가의 여부를 결정

local\_enable=YES

#ftp 전용명령어 중에 write 명령어를 허용할 것인가를 결정

write\_enable=YES

#서버의 FTP 데몬시간을 서버의 표준시간으로 고정할지 설정

use\_localtime=YES

#ftp 접속후에 파일 업로드와 다운로드에 대한 로그를 남길것인가(YES) 남가지 않을 것인가(NO)를 설정

xferlog\_enable=YES

#특정 사용자가 아닌 전체 사용자를 대상으로 chroot()기능을 적용하여

#자기 자신의 홈디렉토링 상위 디렉토리로 이동하지 못하도록 설정

chroot\_local\_user=YES

#FTP 디렉터리에 쓰기 가능하도록 chroot\_local\_user=YES 일때 YES 설정

allow\_writeable\_chroot=YES

#secure chroot()에 사용될 디렉토리를 지정

secure\_chroot\_dir=/var/run/vsftpd/empty

#vsftpd에서 PAM설정파일명으로 사용할 파일명을 지정 /etc/pam.d/vsftpd 파일이 사용

pam\_service\_name=vsftpd

ftpd\_banner=Welcome to My FTP Server!

#SSL을 통한 보안접속을 지원할 것인지에 대해 설정

ssl\_enable=no

rsa\_cert\_file=/etc/ssl/certs/ssl-cert-snakeoil.pem

rsa\_private\_key\_file=/etc/ssl/private/ssl-cert-snakeoil.key

listen\_port=21

pasv\_min\_port=60020

pasv\_max\_port=60030

#utf8\_filesystem=YES

#local\_umask=022

5. sudo systemctl restart vsftpd

- listen=NO로 restart하니 실행이 안되서 YES로 변경하고 나니 실행 잘됨

6. root 계정으로 변환

7. cd ~

8. mkdir firewall\_rules

9. cd firewall\_rules

10. iptables-save > 20210727.rules 현재 날짜를 쓰기

11. iptables-restore < 20210727.rules 방화벽 설정 복구 할 때 사용

12. iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 21 -j ACCEPT

13. iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 60020:60030 -j ACCEPT

14. iptables-save > 20210727.rules

15. 20210727.rules 파일 수정 굵은 글씨 부분 추가

# Generated by iptables-save v1.8.4 on Tue Jul 27 14:20:38 2021

\*filter

:INPUT ACCEPT [4:128]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [3:384]

**-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT**

**-A INPUT -p icmp -j ACCEPT**

**-A INPUT -i lo -j ACCEPT**

-A INPUT -p tcp -m state --state NEW -m tcp --dport 21 -j ACCEPT

-A INPUT -p tcp -m state --state NEW -m tcp --dport 60020:60030 -j ACCEPT

**-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT**

**-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT**

**-A INPUT -j REJECT --reject-with icmp-host-prohibited**

**-A FORWARD -j REJECT --reject-with icmp-host-prohibited**

COMMIT

# Completed on Tue Jul 27 14:20:38 2021

16. 새 터미널에서 firewall\_rules로 이동해서 iptables-restore < 20210727.rules

17. iptables -nL 로 설정 확인

18. apt-get install iptables-persistent 설정 유지를 위해 설치

19. apt-get install iptables-persistent

- yes(엔터) yes(엔터)

/etc/iptables 라는 폴더가 생성되고

rules.v4, rules.v6 가 생성된다.

부팅시 실행되는 파일을 갱신하자. 아래의 작업은 방화벽 수정할 때마다 해주어야 한다.

20. iptables-save > /etc/iptables/rules.v4

Ipv6 차단하기

21. gedit /etc/sysctl.conf

맨 밑에

net.ipv6.conf.all.disable\_ipv6 = 1

net.ipv6.conf.default.disable\_ipv6 = 1

net.ipv6.conf.lo.disable\_ipv6 = 1

추가

22. sysctl -p 설정 파일 적용

23. 검사

cat /proc/sys/net/ipv6/conf/all/disable\_ipv6

1 이면 disable 적용 완료

24. reboot

25. sudo iptables -nL 확인시 설정 남아 있으면 완료

26. 공유기 포트포워딩 설정후 접속 확인

**SSH 설치**

1. sudo apt-get install ssh

2. 루트 계정 접속

3. gedit /etc/ssh/sshd\_config

아래 내용 변경

Port 22 (ssh 포트 번호)

IgnoreRhosts yes (자동 로그인 접근 무시)

PermitRootLogin yes (root 계정 접근 가능)

3. iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT

4. cd /firewall\_rules에서

- iptables-save > 20210727.rules

하고 추가된 설정을

**-A INPUT -j REJECT --reject-with icmp-host-prohibited 위로 바꿔서 저장**

- iptables-resotre < 20210727.rules

- iptables-save > /etc/iptables/rules.v4

- reboot

5. systemctl restart ssh

6. iptime 포트 포워딩 22 설정

7. putty 접속 체크 확인 완료

주의사항 4번 설정을 꼭 정확히 해줘야 외부 접속이 된다.

**JAVA 설치 (jdk-11버전)**

1. sudo apt update

2. sudo apt upgrade

3. sudo apt install openjdk-11-jdk

4. java -version 설치 확인

5. javac -version 설치 확인

6. sudo gedit ~/.bashrc

맨아래에 아래 내용 추가

- export JAVA\_HOME=$(dirname $(dirname $(readlink -f $(which java))))

- export PATH=$PATH:$JAVA\_HOME/bin

7. source ~/.bashrc

8. echo $JAVA\_HOME

/usr/lib/jvm/java-11-openjdk-am64 나오면 성공

**TOMCAT9 설치**

1. sudo apt update

2. sudo apt upgrade

3. sudo apt install tomcat9 tomcat9-admin

4. ss -ltn 8080포트 동작 확인

5. 루트 접속 iptables 8080포트 설정

/firewall\_rules 아래 파일 복사 해서 복사된 파일에

-A INPUT -p tcp -m state --state NEW -m tcp --dport 8080 -j ACCEPT 붙여넣기

- iptables-resotre < 20210728.rules (파일명은 날짜순으로 제가 만든 것)

- iptables-save > /etc/iptables/rules.v4

- reboot

6. 공유기에 8080포트 설정

7. sudo gedit /etc/tomcat9/tomcat-users.xml

하단에

<role rolename="admin-gui"/>

<role rolename="manager-gui"/>

<user username="tomcat" password="비밀번호" roles="admin-gui,admin-script,manager-gui,manager-script, manager-status"/>내용추가

아이디, 비밀번호 란에 ubuntu 계정 아이디 비밀번호 입력

8 passwd tomcat 으로 tomcat password 설정

9. sudo systemctl restart tomcat9

10. sudo systemctl enable tomcat9

11. tomcat 접속해보기 완료

http://localhost:8080

http://공유기ip:8080/manger/html

**MariaDB 설치**

1. sudo apt install mariadb-server mariadb-client

2. sudo mysql\_secure\_installation

- ENTER -> Y -> 새루트 비밀번호 입력 -> 계속Y

3. MariaDB 계정 생성

- sudo mysql -u root -p

- use mysql; (maria db 사용)

- select host, user, password from user; (사용중인 계정 조회)

- create user ‘생성할아이디’@'%' identified by '비밀번호';

- grant all privileges on \*.\* to '생성한아이디’@'%';

4. sudo gedit /etc/mysql/mariadb.conf.d/50-server.cnf

Port 주석 풀고

bind-address 0.0.0.0 으로 수정

5. sudo systemctl restart mariadb

6. 루트 접속 iptables 3306포트 설정

/firewall\_rules 아래 파일 복사 해서 복사된 파일에

-A INPUT -p tcp -m state --state NEW -m tcp --dport 3306 -j ACCEPT 붙여넣기

- iptables-resotre < 20210728.rules (파일명은 날짜순으로 제가 만든 것)

- iptables-save > /etc/iptables/rules.v4

- reboot

7. 공유기에 3306포트 설정

8. 접속해보기 완료

**SVN 설치**

1. sudo apt install subversion

2. /home/hslee 에 svn 폴더 생성

3. sudo svnadmin create svnrepo

4. cd svnrepo/conf

5. sudo gedit ./svnserve.conf

아래 내용 주석 풀고 수정

anon-access = none # 익명 엑세스의 경우 읽기까지 가능

auth-access = write # 권한있는 엑세스의 경우 쓰기까지 가능

password-db = passwd # 사용자 계정 정보 DB로 passwd 파일 사용

authz-db = authz # 사용자 계정 권한 정보 DB로 authz 파일 사용

6. sudo gedit ./passwd

맨 아래

사용할 ID = 사용할 패스워드 입력 후 저장

7. sudo gedit ./authz

맨아래 아래 내용 추가

[/]

\* = rw

hslee=rw

8. sudo svnserve -d -r /home/hslee/svn/svnrepo svn 가동

9. svn 서비스 등록

- sudo gedit /etc/rc.local

아래 내용 저장

#Subversion

svnserve -d -r /home/hslee/svn/svnrepo

- sudo touch /etc/init.d/svnserve

- sudo gedit /etc/init.d/svnserve

아래 내용 저장

#! /bin/sh

### BEGIN INIT INFO

# Provides: svnserve

# Required-Start: $local\_fs $syslog $remote\_fs

# Required-Stop: $local\_fs $syslog $remote\_fs

# Default-Start: 2 3 4 5

# Default-Stop: 0 1 6

# Short-Description: Start svnserve

### END INIT INFO

# Author: Michal Wojciechowski <odyniec@odyniec.net>

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

DESC="svnserve"

NAME=svnserve

DAEMON=/usr/bin/$NAME

DAEMON\_ARGS="-d -r /home/hslee/svn/svnrepo --listen-port 3690"

PIDFILE=/var/run/$NAME.pid

SCRIPTNAME=/etc/init.d/$NAME

[ -x "$DAEMON" ] || exit 0

[ -r /etc/default/$NAME ] && . /etc/default/$NAME

. /lib/init/vars.sh

. /lib/lsb/init-functions

do\_start()

{

start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON --test > /dev/null \

|| return 1

start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $DAEMON -- \

$DAEMON\_ARGS \

|| return 2

}

do\_stop()

{

start-stop-daemon --stop --quiet --retry=TERM/30/KILL/5 --pidfile $PIDFILE --name $NAME

RETVAL="$?"

[ "$RETVAL" = 2 ] && return 2

start-stop-daemon --stop --quiet --oknodo --retry=0/30/KILL/5 --exec $DAEMON

[ "$?" = 2 ] && return 2

rm -f $PIDFILE

return "$RETVAL"

}

case "$1" in

start)

[ "$VERBOSE" != no ] && log\_daemon\_msg "Starting $DESC" "$NAME"

do\_start

case "$?" in

0|1) [ "$VERBOSE" != no ] && log\_end\_msg 0 ;;

2) [ "$VERBOSE" != no ] && log\_end\_msg 1 ;;

esac

;;

stop)

[ "$VERBOSE" != no ] && log\_daemon\_msg "Stopping $DESC" "$NAME"

do\_stop

case "$?" in

0|1) [ "$VERBOSE" != no ] && log\_end\_msg 0 ;;

2) [ "$VERBOSE" != no ] && log\_end\_msg 1 ;;

esac

;;

restart|force-reload)

log\_daemon\_msg "Restarting $DESC" "$NAME"

do\_stop

case "$?" in

0|1)

do\_start

case "$?" in

0) log\_end\_msg 0 ;;

1) log\_end\_msg 1 ;; # Old process is still running

\*) log\_end\_msg 1 ;; # Failed to start

esac

;;

\*)

# Failed to stop

log\_end\_msg 1

;;

esac

;;

\*)

echo "Usage: $SCRIPTNAME {start|stop|restart|force-reload}" >&2

exit 3

;;

esac

exit 0

- sudo chmod u+x /etc/init.d/svnserve

- sudo update-rc.d svnserve defaults

- sudo service svnserve start

10. reboot 후 systemctl status svnserve 되면 성공

11. 루트 접속 iptables 3690포트 설정

/firewall\_rules 아래 파일 복사 해서 복사된 파일에

-A INPUT -p tcp -m state --state NEW -m tcp --dport 3690 -j ACCEPT 붙여넣기

- iptables-resotre < 20210728.rules (파일명은 날짜순으로 제가 만든 것)

- iptables-save > /etc/iptables/rules.v4

- reboot

12. 공유기에 3690포트 설정

13. eclipse에서 svn 연결 해보기 완료

**Eclipse 프로젝트 svn 업로드**

구글링 참조

**Jenkins 설치**

1. wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

2. echo deb http://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list

3. apt update

4. apt -y install jenkins

5. gedit /etc/default/jenkins 포트 8090으로 변경후 systemctl restart jenkins

5. iptables 8090 추가

6. iptime 포트포워딩 8090

7. Jenkins 웹 접속

7. sudo cat /var/lib/jenkins/secrets/initialAdminPassword

이 초기 패스워드로 Jenkins 접속 후 권장 플러그인 설치

8. 아이디 생성 하면 완료

추가 svn 플러그인 설치

추가 maven integration 설치

추가 Deploy to container

9. jenkins관리 Global Tool Configuration jdk & maven설정

Global tool configuration 에서 add JDK 선택

- name : openJDK11

-path: /usr/lib/jvm/java-11-openjdk-amd64

Add maven

-name : maven3.6.3

-Install automatically version 3.6.3 선택

저장

Svn 프로젝트 Jenkins 배포

1. 새로운 item 클릭

2. 프로젝트 명 입력 후 maven project 선택 후 OK

3. 소스코드 관리 부분에서 svn의 주소를 입력

4. 아이디 등록을 위해 Add 버튼을 눌러서 권한 생성

5. Pre Steps 에서 invoke top-level Maven targets 클릭

6. Goal에 clean install 입력

7. Build 에서 pom.xml 입력

8. 빌드 후 조치 에서 deploy war/ear to a container 선택

War/ear files \*\*/\*.war

Tomcat 설치한 버전 추가

9. 저장후 빌드 실패

원인: maven 미설치 및 global tool configuration 설정 안함

Global tool configuration 에서 add JDK 선택

- name : openJDK11

-path: /usr/lib/jvm/java-11-openjdk-amd64

Add maven

-name : maven3.6.3

-path : /usr/share/maven

JAVA\_HOME should point to a JDK not a JRE 에러

원인 JDK path 오타 -> 설정 완료

Build는 성공했으나 아래와 같은 에러 발생

추측: java 버전 및 tomcat 버전이 다르고 spring대신 springboot pom.xml 설정이 다름

ERROR: Step ‘Deploy war/ear to a container’ aborted due to exception:

java.lang.InterruptedException: [DeployPublisher][WARN] No wars found. Deploy aborted. %n

변경사항 1. Pom.xml

- java version수정

- <packaging>war</packaging> 추가

<!-- <maven-jar-plugin.version>3.1.1</maven-jar-plugin.version> --> 주석 처리

Jar를war로 변경하는 것 성공 BUT 에러 발생

톰캣 접속 허용이 안되서 찾아보니

/etc/tomcat9/Catalina/localhost 아래 host-manager.xml, manager.xml에 외부접속허용 설정해야함

그래도 안되서 보니 tomcat-user.xml 에 허용 하는 것을 더 해줘야함

다른 에러 발생

/var/lib/tomcat9/conf context.xml에

<Context antiResourceLocking="true" antiJARLocking="true"> Context 태그 뒤에 내용추가 저장

배포 안된 이유

Tomcat user 설정으로 tomcat-users.xml 설정해줘야하고 springboot war파일 변환하는 법 사용해야함 정리는 내일

Iptables 설정 관련 내용

https://sata.kr/entry/IPTables-4-m-%EC%98%B5%EC%85%98%EC%9D%98-%ED%99%9C%EC%9A%A9-%EC%98%88%EC%8B%9C