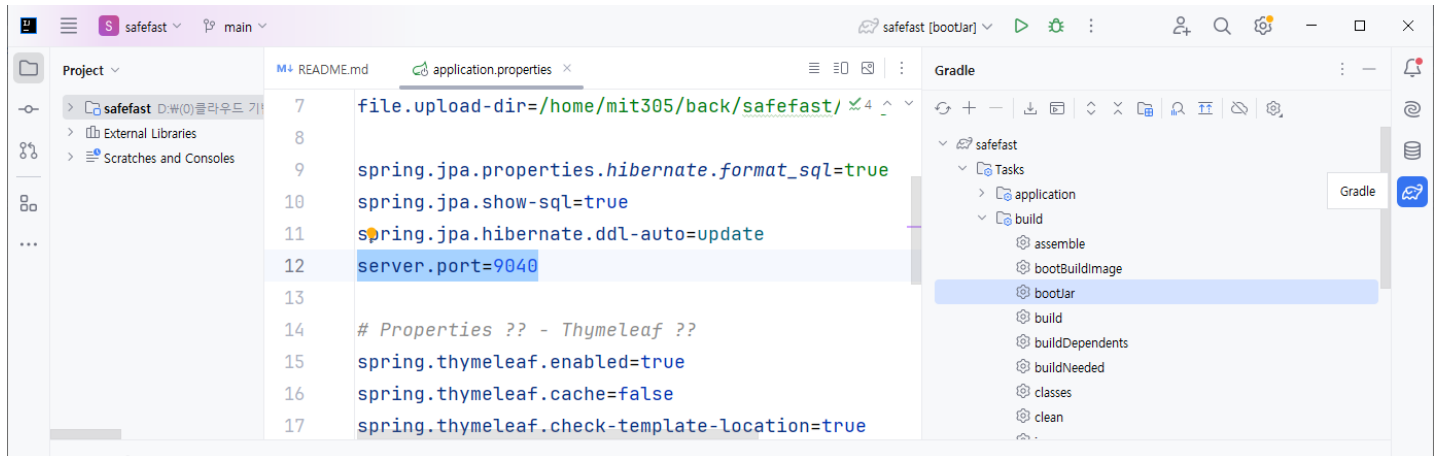


## \*\* IntelliJ에서 JAR 파일 생성



- application.properties 에 **서버포트** 설정
- Gradle 창에서 **boot.jar** 실행
- 프로젝트 폴더 내에 **build>libs** 에서 **JAR 파일 확인** 가능. (이름 변경 후 배포)
-

## **\*\* 서버 배포 준비**

### **- SW 설치**

#모든 프로그램을 최신 버전으로 만들어줘야 한다.

```
$ apt update  
$ apt upgrade
```

#### **# JDK17 설치**

```
$ apt list openjdk-17*           # jdk17 설치 파일 여부 확인  
$ apt install openjdk-17*       # jdk17 설치  
  
# 버전 확인  
$ java -version  
$ javac -version
```

#### **# 구동기 설치**

```
$ apt install apache2  
$ apt install tomcat10          #Tomcat10.1.16
```

- ➔ 구동기를 설치했다면 정상적으로 작동하는지 확인하기 위해
- ➔ 브라우저에 localhost:20, localhost:8080
  
- ➔ 20(Apache), 8080(Tomcat) 포트로 연결해서 info 페이지가 뜬다면 잘 설치한 것이다.

#### **# 서버 관리 프로그램 설치**

```
$ apt install ssh  
$ apt install putty
```

## **\*\* databae 설치 : MariaDB (root: 1234)**

# MariaDB 서버 설치

```
$ sudo apt install mariadb-server
```

# MariaDB 클라이언트 설치

```
$ sudo apt install mariadb-client
```

# MariaDB secure 설정

```
$ sudo mysql_secure_installation
```

root 비밀번호를 입력한다.

root 접속을 위한 비밀번호를 입력하세요.

Enter current password for root (enter for none):

위에서 비밀번호를 설정했기 때문에 y 해서 바꾸거나 같은 비밀번호를 적어도,  
n 을 하여 설정을 안 해도 무방하다.

root 비밀번호를 변경하시겠습니까?

Change the root password? [Y/n]

아래 항목들은 취향 따라 설정하면 된다.

anonymous 계정을 지우시겠습니까?

Remove anonymous users? [Y/n]

test db 를 지우시겠습니까?

Remove test database and access to it? [Y/n]

바로 적용하시겠습니까?

Reload privilege tables now?

설치 되었다면 아래와 같은 문구가 나온다.

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

## **\*\* 기본적인 mariadb 명령어**

# MariaDB 활성화

\$ systemctl enable mariadb

# MariaDB 시작

\$ systemctl start mariadb

# MariaDB 상태 확인

\$ systemctl status mariadb

# 실행중지

Ctrl+c , :q

# MariaDB 중지

\$ sudo systemctl stop mysql

# MariaDB 재시작

\$ sudo systemctl restart mysql

#systemctl 를 못 쓰는 상황이라면 service 명령어를 사용하면 된다

\$ sudo service mariadb start

명령어를 친 후, 비밀번호를 치면 root 계정으로 접속 가능하다.

\$ sudo mysql -u root -p

Enter password:

**\* DB 생성, User 추가, 권한 설정 등 명령어**

# mariadb 접속

\$ sudo mariadb

## # DB 만들기

MariaDB [(none)]> CREATE DATABASE 데이터베이스명;

# DB 확인하기

MariaDB [(none)]> SHOW DATABASES;

# DB 삭제하기

MariaDB [(none)]> DROP DATABASE 데이터베이스명;

#사용자 생성

## 로컬접속용

MariaDB [(none)]> CREATE USER '사용자명'@'localhost' IDENTIFIED BY '비밀번호';

## 외부접속용

MariaDB [(none)]> CREATE USER '사용자명'@'%' IDENTIFIED BY '비밀번호';

# 사용자 확인

MariaDB [(none)]> SELECT HOST, USER, PASSWORD FROM USER;

#사용자 삭제

## 로컬접속용

MariaDB [(none)]> DROP USER '사용자명'@'localhost';

## 외부접속용

MariaDB [(none)]> DROP USER '사용자명'@'%';

# 권한 부여

## 로컬접속용

MariaDB [(none)]> GRANT ALL PRIVILEGES ON 데이터베이스명.\* TO '사용자명'@'localhost';

## 외부접속용

MariaDB [(none)]> GRANT ALL PRIVILEGES ON 데이터베이스명.\* TO '사용자명'@'%';

# 갱신

MariaDB [(none)]> FLUSH PRIVILEGES;

# 접속종료

MariaDB [(none)]> exit

**\*\* 윈도우에서 HeidiSQL 사용하여 SERVER DB 접속 가능.**

# MariaDB 설정

# 외부에서 해당 계정(현 root)으로 접속할 수 있도록 계정에 권한 부여

MariaDB [(none)]> GRANT ALL PRIVILEGES ON \*.\* to 'root'@'%' IDENTIFIED BY '1234';

#FLUSH PRIVILEGES; 명령어로 반영

MariaDB [(none)]> FLUSH PRIVILEGES;

# 원격 접속 대상에 접속 설정

\$ sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

```
GNU nano 7.2 /etc/mysql/mariadb.conf.d/50-server.cnf
basedir                = /usr
#datadir                = /var/lib/mysql
#tmpdir                 = /tmp

# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address           = 127.0.0.1
port = 3306
bind-address            = 0.0.0.0

#
# * Fine Tuning
#

#key_buffer_size        = 128M
#max_allowed_packet     = 1G

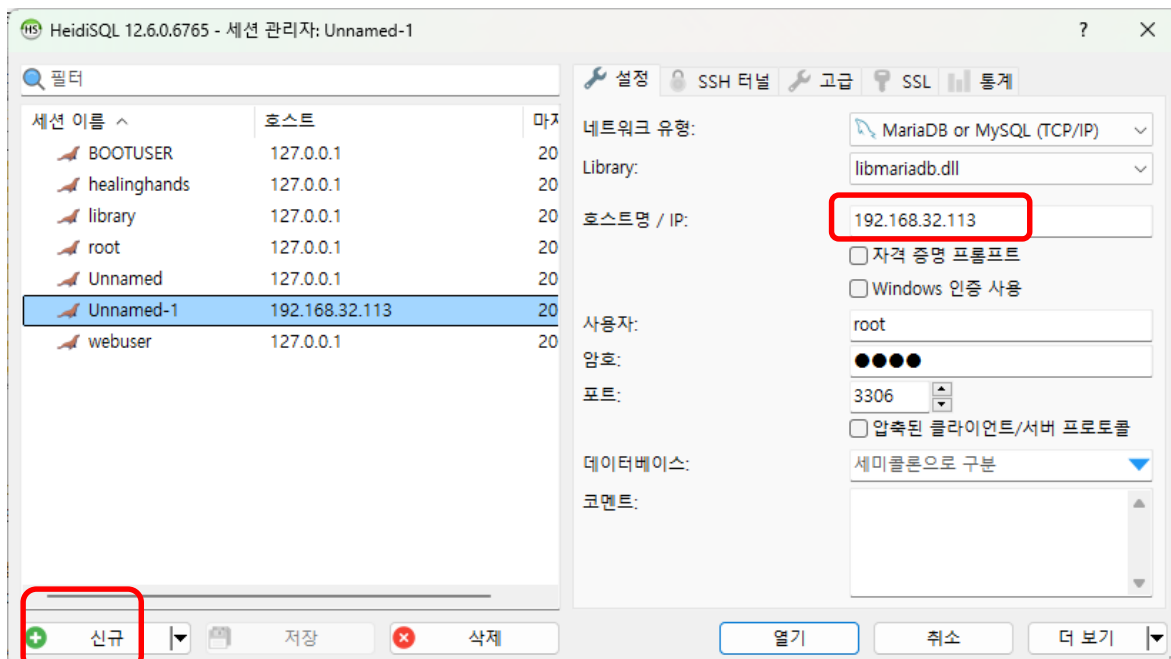
^G Help      ^O Write Out ^W Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace    ^U Paste      ^J Justify   ^/_ Go To Line
```

위와 같이 내용 추가 및 수정

# 포트 방화벽 해제

\$ sudo ufw allow 3306

\$ sudo systemctl restart mariadb

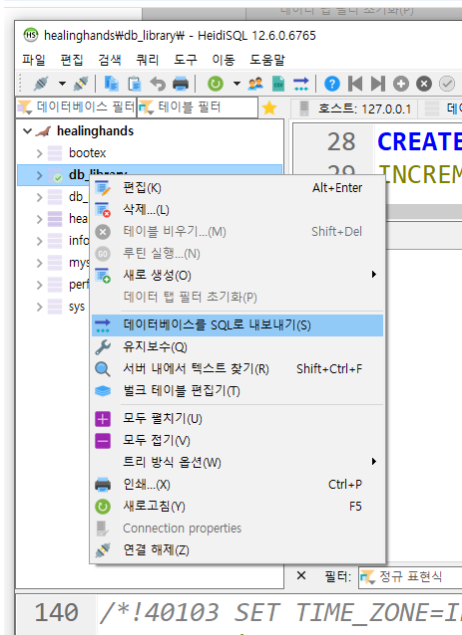


## \*\* HeidiSQL을 활용한 데이터베이스 이관

### 1. HeidiSQL에서 DB 내보내기

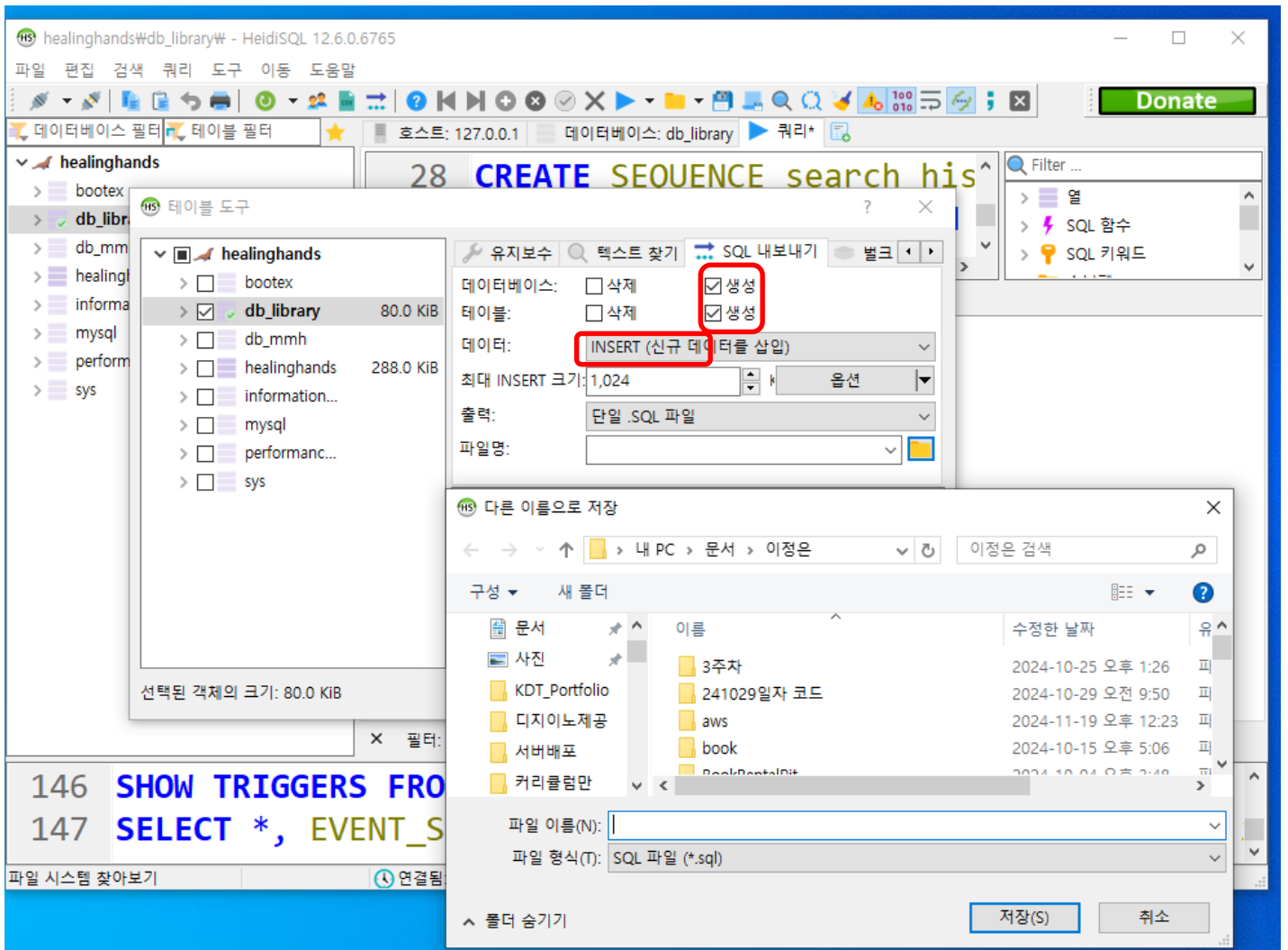
#### 1) SQL로 내보내기

HeidiSQL을 통해 데이터베이스에 접속한 후 내보낼 DB를 우클릭하여 " 선택



#### 2) 내보낼 요소 및 경로 설정

데이터베이스, 테이블 중 SQL로 내보내기가 필요한 요소만 생성에 체크하고 경로를 지정하여 내보내기 클릭. 이 때 데이터까지 함께 옮기기 위해 **INSERT를 선택**.



### 3) 파일 확인

해당 경로에 \*.SQL파일이 생성되었다면 완료.

## 2. HeidiSQL에서 DB 가져오기

### 1) 이관 받을 DB접속

작성자는 현재 기존 서버에서 다른 서버로 데이터를 이관시켜야 하므로 해당 서버로 HeidiSQL 접속

#### 2-1) 쿼리로 불러와서 실행하기

파일>SQL 파일 불러오기 > 불러올 파일 선택

쿼리탭에 기존 데이터베이스를 구성하는 쿼리가 생성.

우클릭>실행으로 전체 적용할 수 있으며 수정하여 적용 가능.

#### 2-2) SQL파일 실행

파일>SQL파일 실행 메뉴로 즉시 실행 가능. 적용 전 수정이 불가능.

※ 데이터베이스 복원 후에는 새로고침(F5)!



## \*\* 멀티 배포를 위한 설정

```
root@mithrd305:# cd /opt/tomcat/apache-tomcat-9.0.89/conf
```

```
root@mithrd305:/opt/tomcat/apache-tomcat-9.0.89/conf# ls
```

```
Catalina    Catalina4    catalina.properties  jaspic-providers.xsd  tomcat-users.xml
Catalina2   Catalina5    context.xml          logging.properties    tomcat-users.xsd
Catalina3   catalina.policy  jaspic-providers.xml  server.xml            web.xml
```

```
root@mithrd305:/opt/tomcat/apache-tomcat-9.0.89/conf# vi server.xml    # vi 에디터.pdf
```

```
<Service name="Catalina">

  <Connector port="8080" protocol="HTTP/1.1"

    connectionTimeout="20000"

    redirectPort="8443"

    maxParameterCount="1000"

    maxThreads="250"

    maxHttpHeaderSize="8192"

    emptySessionPath="true"

    enableLookups="false"

    acceptCount="100"

    disableUploadTimeout="true"

  />

  <Connector protocol="AJP/1.3"

    address="::1"

    port="8009"

    redirectPort="8443"

    maxParameterCount="1000"

  />

  <Host name="localhost"  appBase="/home/mit305/back"

    unpackWARs="true" autoDeploy="true">

    <Context path="" docBase="" reloadable="true"/>

  </Host>

</Service>
```

## \*\* 배포

sftp://mitstudent01@192.168.35.8 - FileZilla

파일(F) 편집(E) 보기(V) 전송(T) 서버(S) 북마크(B) 도움말(H)

호스트(H): sftp://192.168.35.8 사용자명(U): mitstudent01 비밀번호(W): ●●●●●●●●

상태: Listing directory /home/mit305/back/sendbox  
상태: "/home/mit305/back/sendbox" 디렉터리 목록 조회 성공

로컬 사이트: omw 리모트 사이트: /home/mit305/back/sendbox

파일명 ^

파일명	크기	파일 ...	최종 수...	권한	소유...
..					
.android					
.cache					
.eclipse					
.gradle					
.jdk					
.lemminx					
nohup.out	723,...	OUT ...	2024-0...	-rw--...	root r...
sendbox.jar	78,3...	Exec...	2024-0...	-rw-r...	mitstu...

7 파일 및 38 디렉터리 2 파일. 총 크기: 79,073,617 바이트

서버/로컬 파일    방향    리모트 파일    크기    우...    상태

대기 파일    전송 실패    전송 성공

대기열: 비었음

192.168.35.8 - PuTTY

```
login as: mitstudent01
mitstudent01@192.168.35.8's password: █
```

## \* cd 명령으로 각 프로젝트의 카테고리로 이동

### ● Jar 또는 war 파일 실행

\$ java -jar {jar 파일명}.jar

\$ java -jar {war 파일명}.war

예시>

mitstudent01@mithrd305:~\$ cd sandbox

mitstudent01@mithrd305:~/sandbox \$ ls

mitstudent01@mithrd305:~/sandbox \$ java -jar sandbox.jar

```
root@mithrd305: ~
mitstudent01@mithrd305:~/exeasy$ java -jar exeasy.war

:: Spring Boot ::
(v3.2.4)

2024-06-04T16:10:55.887+09:00 INFO 60458 --- [MakeBoots] [main] com.example.demo.MakeBootsApplication : Starting
MakeBootsApplication v0.0.1-SNAPSHOT using Java 17.0.11 with PID 60458 (/home/mit305/back/exeas...
2024-06-04T16:10:55.889+09:00 INFO 60458 --- [MakeBoots] [main] com.example.demo.MakeBootsApplication : No active
profile set, falling back to 1 default profile: "default"
2024-06-04T16:10:56.441+09:00 INFO 60458 --- [MakeBoots] [main] .s.d.r.c.RepositoryConfigurationDelegate : Multiple
Spring Data modules found, entering strict repository configuration mode
2024-06-04T16:10:56.442+09:00 INFO 60458 --- [MakeBoots] [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrap
ping Spring Data JDBC repositories in DEFAULT mode.
2024-06-04T16:10:56.457+09:00 INFO 60458 --- [MakeBoots] [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished
Spring Data repository scanning in 11 ms. Found 0 JDBC repository interfaces.
2024-06-04T16:10:56.465+09:00 INFO 60458 --- [MakeBoots] [main] .s.d.r.c.RepositoryConfigurationDelegate : Multiple
Spring Data modules found, entering strict repository configuration mode
2024-06-04T16:10:56.466+09:00 INFO 60458 --- [MakeBoots] [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrap
ping Spring Data JPA repositories in DEFAULT mode.
2024-06-04T16:10:56.471+09:00 INFO 60458 --- [MakeBoots] [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished
Spring Data repository scanning in 2 ms. Found 0 JPA repository interfaces.
2024-06-04T16:10:56.834+09:00 INFO 60458 --- [MakeBoots] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat in
itialized with port 9080 (http)
2024-06-04T16:10:56.842+09:00 INFO 60458 --- [MakeBoots] [main] o.apache.catalina.core.StandardService : Starting
service [Tomcat]
2024-06-04T16:10:56.842+09:00 INFO 60458 --- [MakeBoots] [main] o.apache.catalina.core.StandardEngine : Starting
Servlet engine: [Apache Tomcat/10.1.19]
2024-06-04T16:10:57.650+09:00 INFO 60458 --- [MakeBoots] [main] org.apache.jasper.servlet.TldScanner : At least
one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were s
canned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.
2024-06-04T16:10:58.028+09:00 INFO 60458 --- [MakeBoots] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializ
ing Spring embedded WebApplicationContext
2024-06-04T16:10:58.028+09:00 INFO 60458 --- [MakeBoots] [main] w.s.c.ServletWebServerApplicationContext : Root WebA
pplicationContext: initialization completed in 2076 ms
2024-06-04T16:10:58.132+09:00 INFO 60458 --- [MakeBoots] [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204
: Processing PersistenceUnitInfo [name: default]
2024-06-04T16:10:58.173+09:00 INFO 60458 --- [MakeBoots] [main] org.hibernate.Version : HHH000412
: Hibernate ORM core version 6.4.4.Final
2024-06-04T16:10:58.194+09:00 INFO 60458 --- [MakeBoots] [main] o.h.c.internal.RegionFactoryInitiator : HHH000026
: Second-level cache disabled
2024-06-04T16:10:58.327+09:00 INFO 60458 --- [MakeBoots] [main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTi
meWeaver setup: ignoring JPA class transformer
2024-06-04T16:10:58.344+09:00 INFO 60458 --- [MakeBoots] [main] com.zaxxer.hikari.HikariDataSource : HikariPoo
l-1 - Starting...
2024-06-04T16:10:58.405+09:00 INFO 60458 --- [MakeBoots] [main] com.zaxxer.hikari.pool.HikariPool : HikariPoo
l-1 - Added connection org.mariadb.jdbc.Connection@559d19c
```

## \*\* 종료

- Ctrl+c 로 실행명령 종료.

### 종료 후 재 실행 후 포트 충돌 에러 발생 시

- ps -ef 명령어를 실행해 현재 실행중인 프로세스 중 java 명령어 찾기.

- kill -9 {ProcessID(PID)}를 실행.

```
root@mithrd305: /opt/tomcat/apache-tomcat-9.0.89/conf
mit305 22843 3143 0 May30 ? 00:00:00 /snap/firefox/4336/usr/lib/firefox/firefox -contentproc -childID 105 -isForB
mit305 22906 3143 0 May30 ? 00:00:00 /snap/firefox/4336/usr/lib/firefox/firefox -contentproc -childID 106 -isForB
mit305 22910 3143 0 May30 ? 00:00:00 /snap/firefox/4336/usr/lib/firefox/firefox -contentproc -childID 107 -isForB
mit305 22950 3143 0 May30 ? 00:00:00 /snap/firefox/4336/usr/lib/firefox/firefox -contentproc -childID 108 -isForB
mit305 23762 3143 0 May30 ? 00:07:00 /snap/firefox/4336/usr/lib/firefox/firefox -contentproc -childID 112 -isForB
mit305 24012 3143 0 May30 ? 00:00:00 /snap/firefox/4336/usr/lib/firefox/firefox -contentproc -childID 115 -isForB
root 24666 2130 0 May30 ? 00:13:57 java -jar sendbox.jar
mit305 25075 2342 0 May30 ? 00:00:00 /usr/libexec/gvfsd-network --spawner :1.20 /org/gtk/gvfs/exec_spaw/4
mit305 25097 2342 0 May30 ? 00:00:00 /usr/libexec/gvfsd-dnssd --spawner :1.20 /org/gtk/gvfs/exec_spaw/6
mit305 26224 3143 0 May30 ? 00:00:00 /snap/firefox/4336/usr/lib/firefox/firefox -contentproc -childID 116 -isForB
mit305 26272 3143 0 May30 ? 00:00:00 /snap/firefox/4336/usr/lib/firefox/firefox -contentproc -childID 117 -isForB
root 27372 1 0 May30 ? 00:00:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data 27373 27372 0 May30 ? 00:00:00 nginx: worker process
www-data 27374 27372 0 May30 ? 00:00:00 nginx: worker process
www-data 27375 27372 0 May30 ? 00:00:00 nginx: worker process
www-data 27376 27372 0 May30 ? 00:00:00 nginx: worker process
www-data 27378 27372 0 May30 ? 00:00:00 nginx: worker process
www-data 27379 27372 0 May30 ? 00:00:00 nginx: worker process
www-data 27380 27372 0 May30 ? 00:00:00 nginx: worker process
www-data 27381 27372 0 May30 ? 00:00:00 nginx: worker process
```

## \*\* 백그라운드에서도 실행하기!

nohup + & 명령어로 우분투 터미널의 세션 연결이 종료됐을 때도 jar 파일을 멈추지 않고 실행할 수 있다.

**nohup** java -jar {jar 파일명}.jar **&**