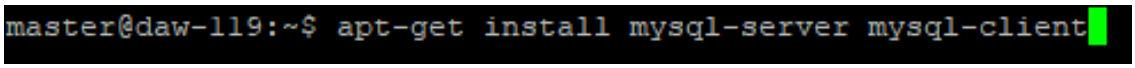
Subir la aplicacion a un servidor

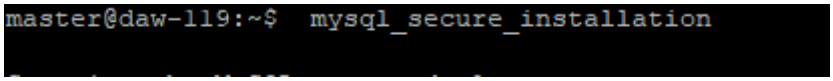
- Instalamos mysql server y mysql client

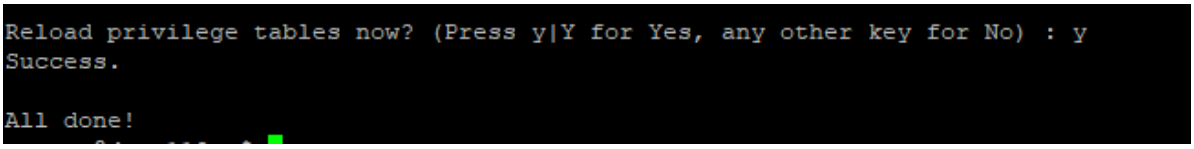
```
apt-get install mysql-server mysql-client
```




- Creamos la instalacion segura de mysql

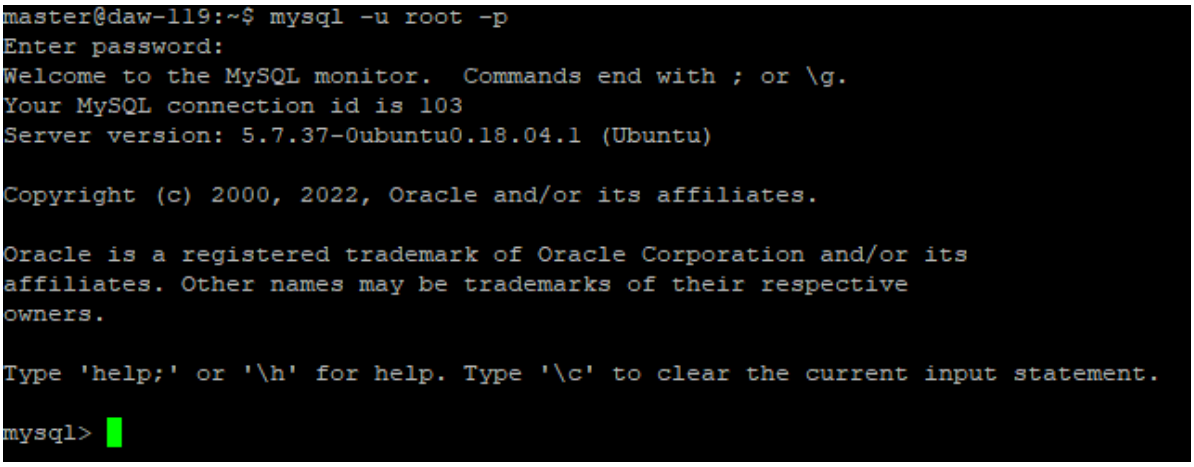
```
mysql_secure_instalation
```



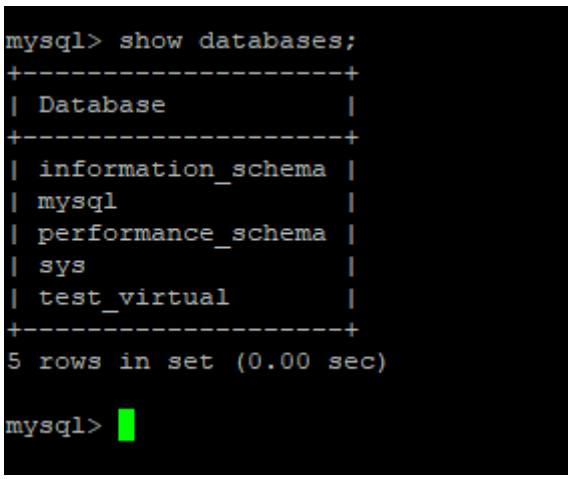

- Entramos como root

```
mysql -u root -p
```



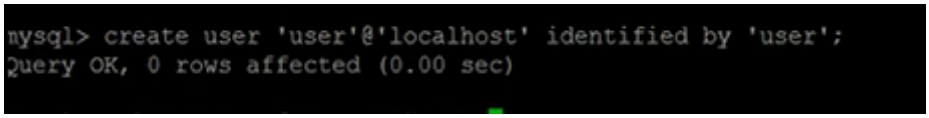

- Vemos las bases de datos creadas en el sistema y creamos la BBDD test_virtual

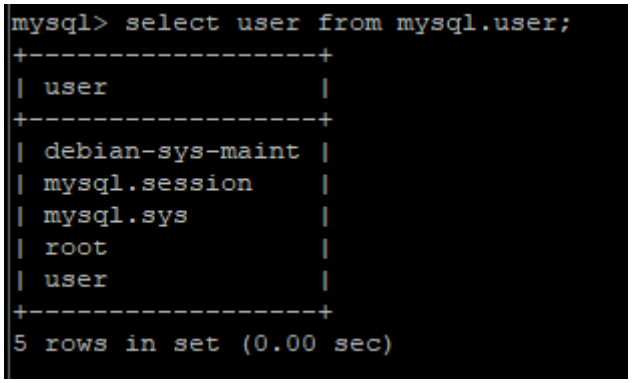
```
show databases;
create database test_virtual;
```




- Creamos el usuario 'user'

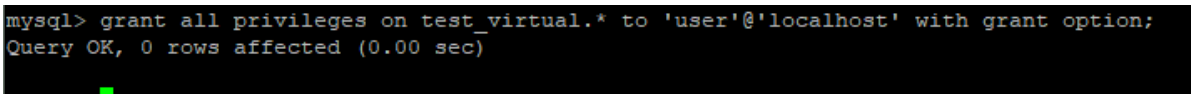
```
create user 'user'@'localhost' identified by 'user';
```




- Damos privilegios al usuario sobre esta tabla

```
grant all privileges on test_virtual.* to 'user'@'localhost' with grant option;
```




- Nos logueamos como usuario para ver la base de datos.

```
mysql -u user -p
```

```
master@daw-119:~$ sudo mysql -u user -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 104
Server version: 5.7.37-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| test_virtual       |
+--------------------+
2 rows in set (0.00 sec)
```

- Instalamos Java, hay que instalar maven previamente.

```
apt-get install maven
```

```
master@daw-119:~$ apt-get install maven
```

- Instalamos jpk8

```
apt-get install openjdk-8-jdk
```

```
master@daw-119:~$ sudo apt-get install openjdk-8jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

- Comprobamos la versión de Java actual.

```
java -version
```

```
master@daw-119:~$ java -version
openjdk version "11.0.13" 2021-10-19
OpenJDK Runtime Environment (build 11.0.13+8-Ubuntu-0ubuntu1.18.04)
OpenJDK 64-Bit Server VM (build 11.0.13+8-Ubuntu-0ubuntu1.18.04, mixed mode)
```

- Instalamos el servidor ftp

```
apt-get install vsftpd
```

```
master@daw-119:~$ sudo apt-get install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

- Editamos el archivo de configuración vsftpd y habilitamos la escritura del local user

```
nano /etc/vsftpd.conf
```

```
 GNU nano 2.9.3                          /etc/vsftpd.conf

# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=NO
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO

# Uncomment this to allow local users to log in.
local_enable=YES
write_enable=YES
#
# Uncomment this to enable any form of FTP write command.
#write_enable=YES
#
```

- Reiniciamos el servicio sftpd.

```
systemctl restart vsftpd.service
```

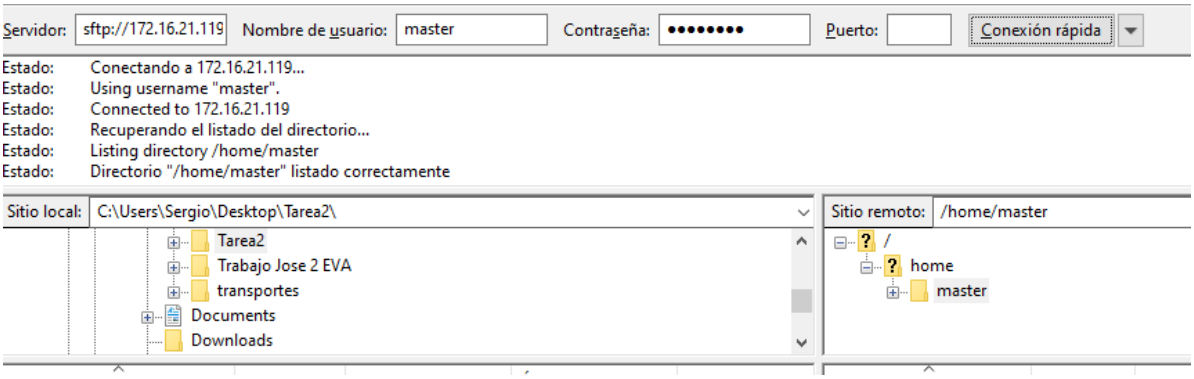
```
master@daw-119:~$ sudo systemctl restart vsftpd.service
```

- Comprobamos que esté habilitado.

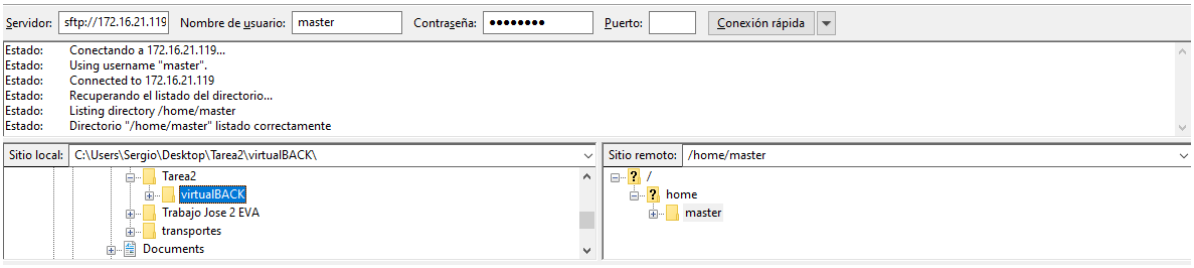
```
systemctl is-enabled vsftpd.service
```

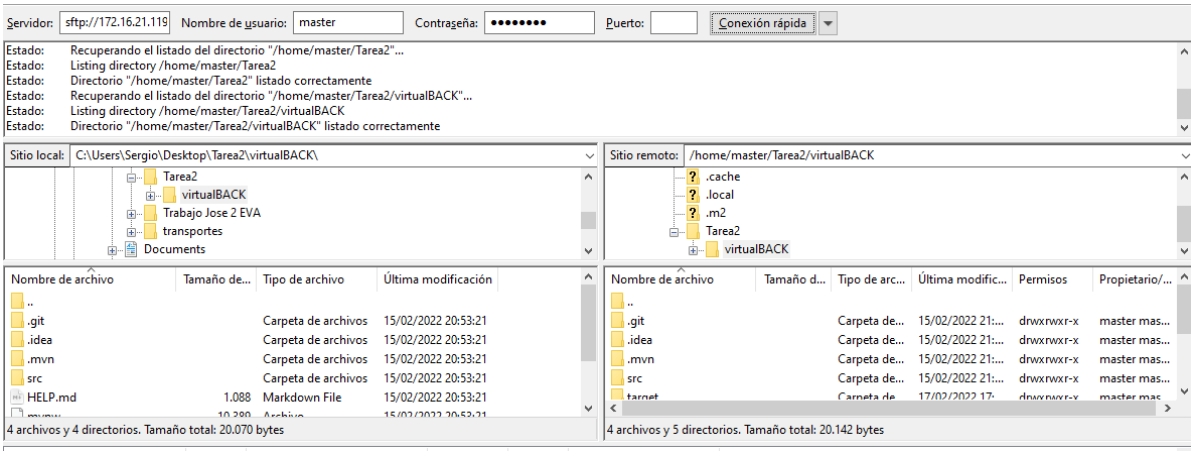
```
master@daw-119:~$ sudo systemctl is-enabled vsftpd.service
enabled
```

- Arrancamos Filezilla y nos conectamos con nuestra ip.




- Subimos nuestro trabajo a nuestra carpeta /home/master

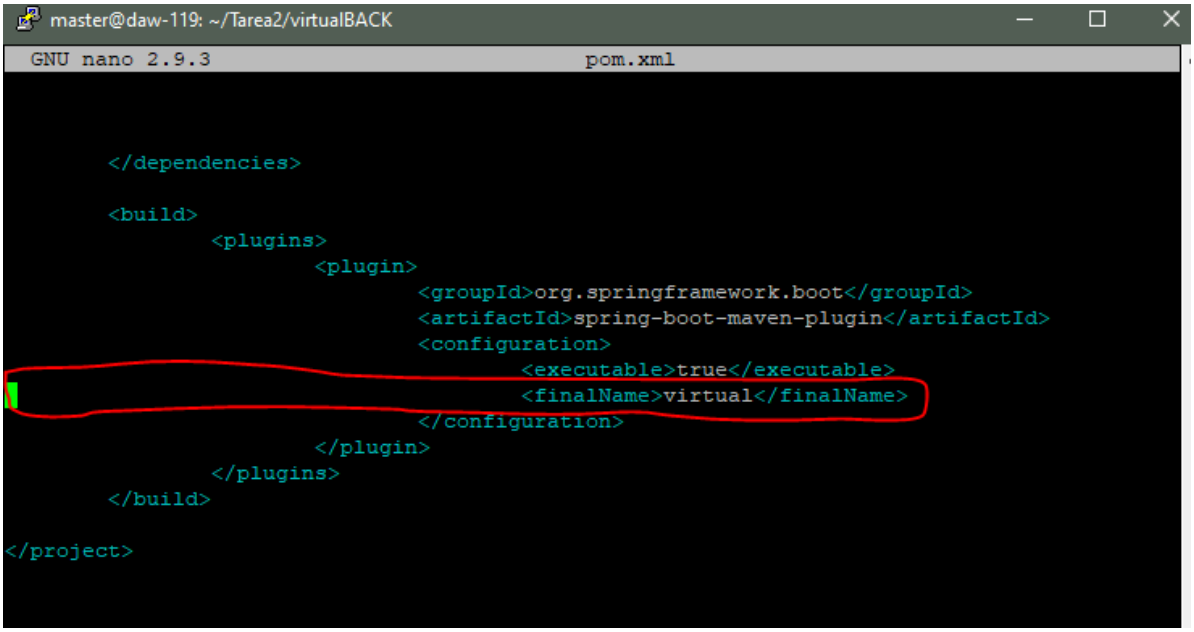





- Accedemos a la ruta donde hemos subido nuestro trabajo.

```
master@daw-119:~$ cd Tarea2/virtualBACK/
master@daw-119:~/Tarea2/virtualBACK$ ls
HELP.md  mvnw  mvnw.cmd  pom.xml  src  target
```

- Editamos el fichero pom.xml




- Instalamos maven

```
mvn install
```

```
[INFO] Installing /home/master/Tarea2/virtualBACK/target/virtual.jar to /root/.m2/repos
y/com/cavanosa/virtual/0.0.1-SNAPSHOT/virtual-0.0.1-SNAPSHOT.jar
[INFO] Installing /home/master/Tarea2/virtualBACK/pom.xml to /root/.m2/repository/com/c
osa/virtual/0.0.1-SNAPSHOT/virtual-0.0.1-SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  44.179 s
[INFO] Finished at: 2022-02-22T20:10:00Z
[INFO] ------------------------------------------------------------------------
```

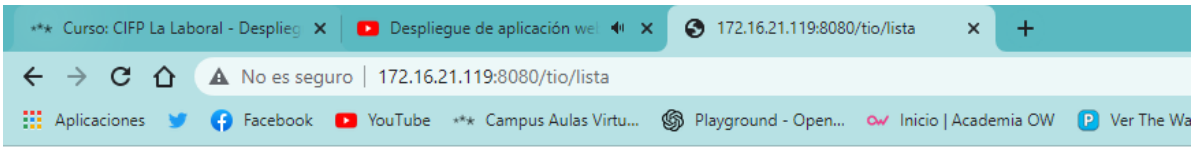
- Ejecutamos para ver si funciona

```
java -jar target/virtual-jar
```

```
master@daw-119:~/Tarea2/virtualBACK$ java -jar target/virtual.jar

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.1.RELEASE)
```

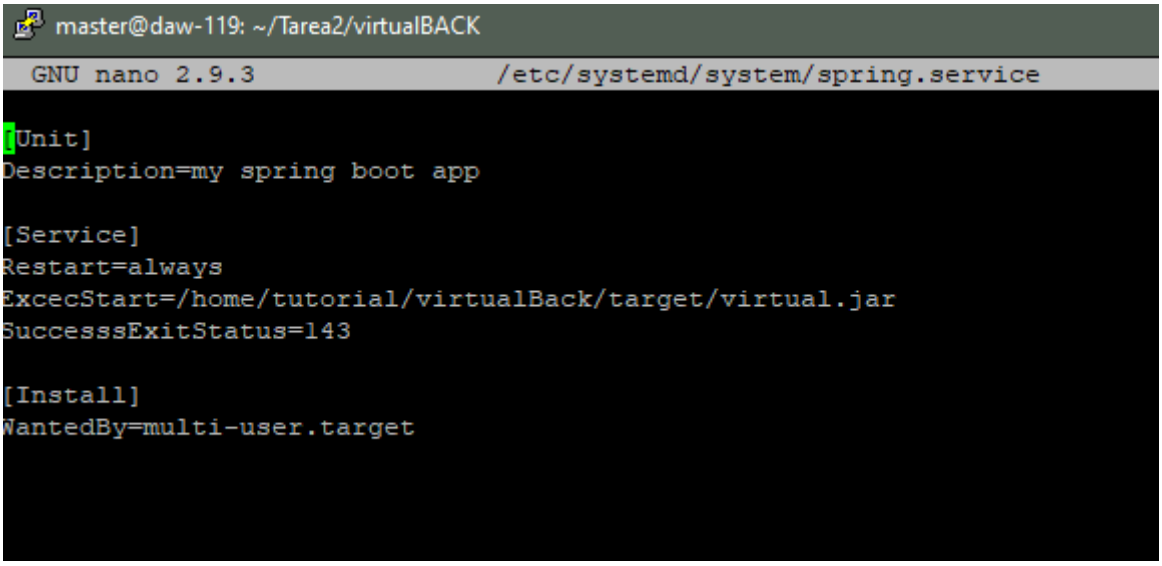
- Desde el navegador, accedemos con nuestra IP y el puerto 8080 para ver que todo funciona correctamente




```
[]
```

- Creamos un servicio para que, cuando el servidor arranque, arranque el servicio de Srping.

Le añadimos las siguientes instrucciones:




- Comprobamos que el servicio de Spring esté habilitado.

```
systemctl is-enabled spring.service
```

```
master@daw-119:~/Tarea2/virtualBACK$ sudo systemctl is-enabled spring.service
enabled
```