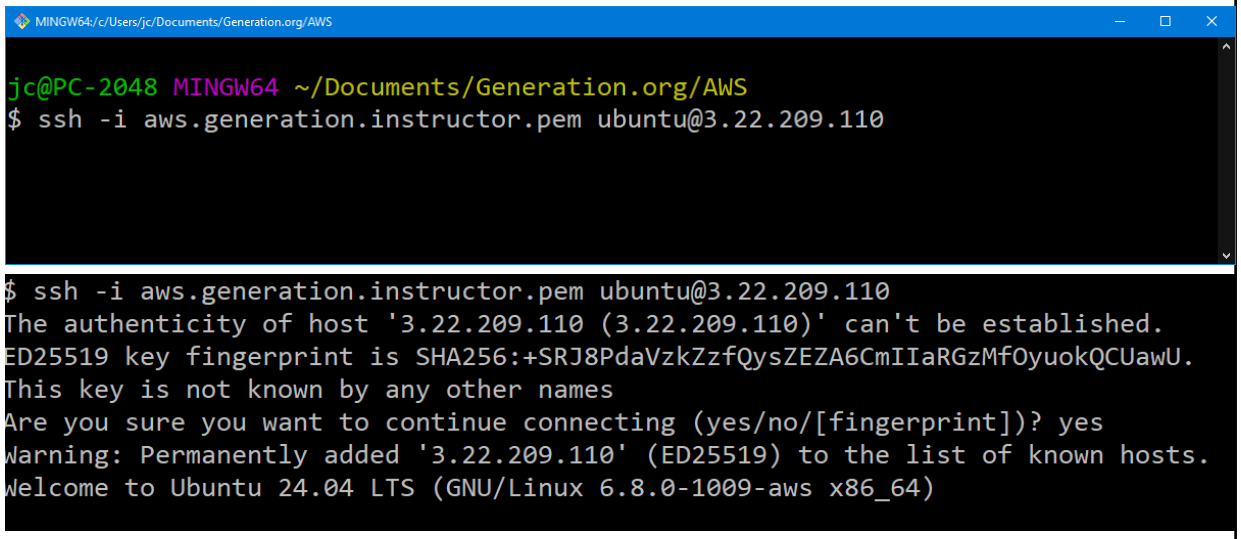
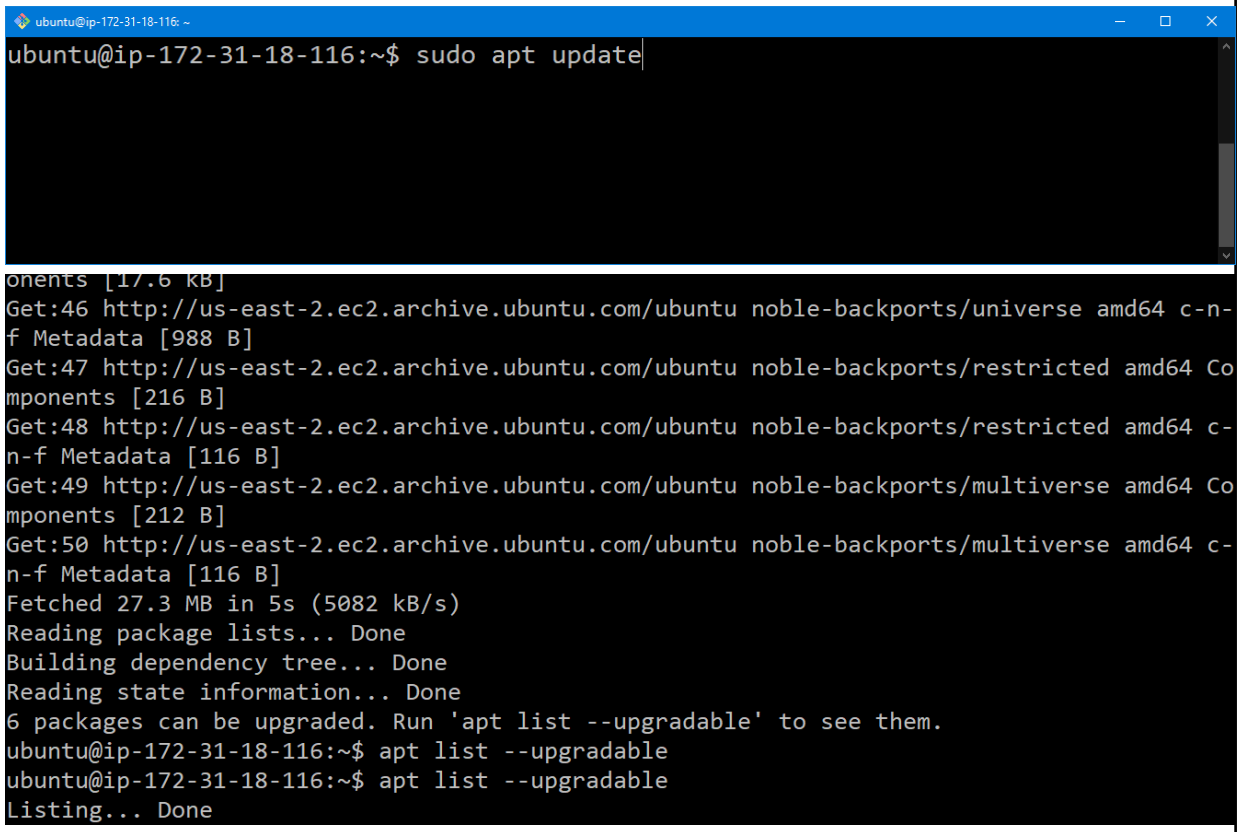
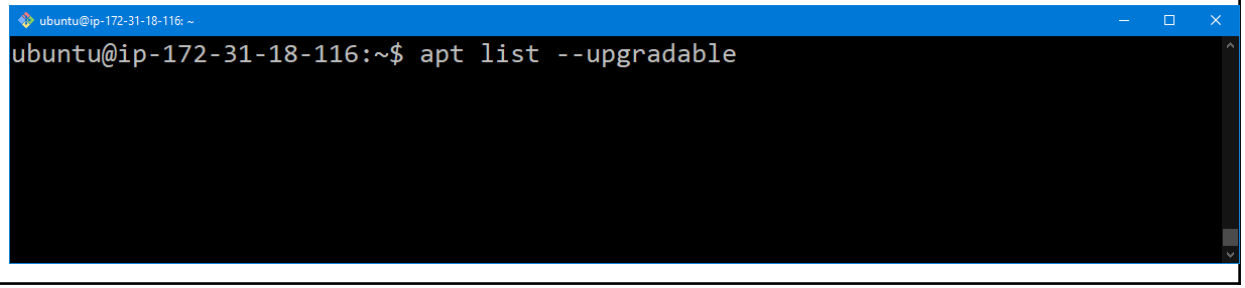
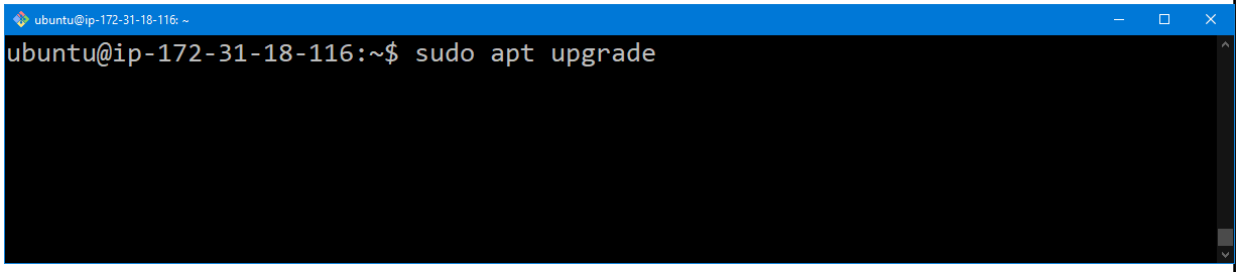
Deploy en AWS

Requisitos:
1. IP Pública del servidor
2. Archivo de la Llave SSH

1.Conectarse al servidor **ssh -i archivollave.pem ubuntu@direccionIP** Si les pregunta si desean continuar, escribir **yes** ***Recomendación** Abrir el git bash en el directorio donde tienen el archivo de la llave	

Pedir la actualización del servidor **sudo apt update**	
Mostrar la lista de paquetes actualizables **apt list --upgradable**	

	ubuntu@ip-172-31-18-116:~$ apt list --upgradable Listing... Done linux-aws/noble-updates,noble-security 6.8.0-1010.10 amd64 [upgradable from: 6.8.0-1009.9] linux-headers-aws/noble-updates,noble-security 6.8.0-1010.10 amd64 [upgradable from: 6.8.0-1009.9] linux-image-aws/noble-updates,noble-security 6.8.0-1010.10 amd64 [upgradable from: 6.8.0-1009.9] openssh-client/noble-security 1:9.6p1-3ubuntu13.4 amd64 [upgradable from: 1:9.6p1-3ubuntu13.3] openssh-server/noble-security 1:9.6p1-3ubuntu13.4 amd64 [upgradable from: 1:9.6p1-3ubuntu13.3] openssh-sftp-server/noble-security 1:9.6p1-3ubuntu13.4 amd64 [upgradable from: 1:9.6p1-3ubuntu13.3]
Actualizar el sistema **sudo apt upgrade** En caso de preguntar si deseas actualizar escribimos **Y**	

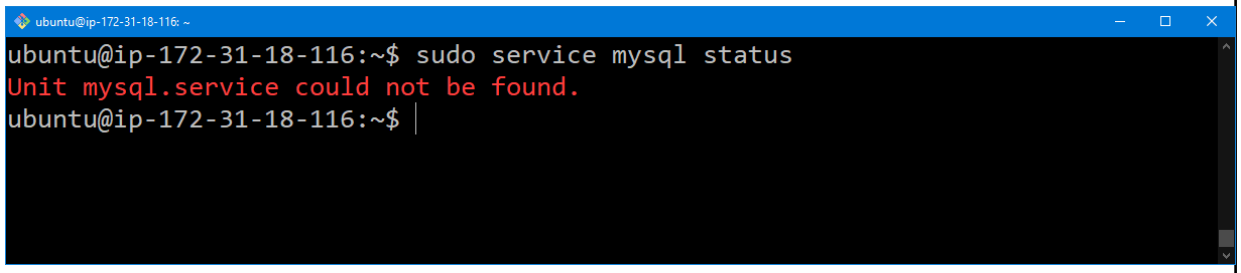
```
ubuntu@ip-172-31-18-116:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  linux-aws-headers-6.8.0-1010 linux-aws-tools-6.8.0-1010 linux-headers-6.8.0-1010-aws
  linux-image-6.8.0-1010-aws linux-modules-6.8.0-1010-aws linux-tools-6.8.0-1010-aws
The following packages will be upgraded:
  linux-aws linux-headers-aws linux-image-aws openssh-client openssh-server
  openssh-sftp-server
6 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
6 standard LTS security updates
Need to get 73.7 MB of archives.
After this operation, 182 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
Restarting the system to load the new kernel will not be handled automatically, so you
should consider rebooting.

No services need to be restarted.

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #3: sshd[1006]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-18-116:~$
```

Instalación de MySQL/MariaDB

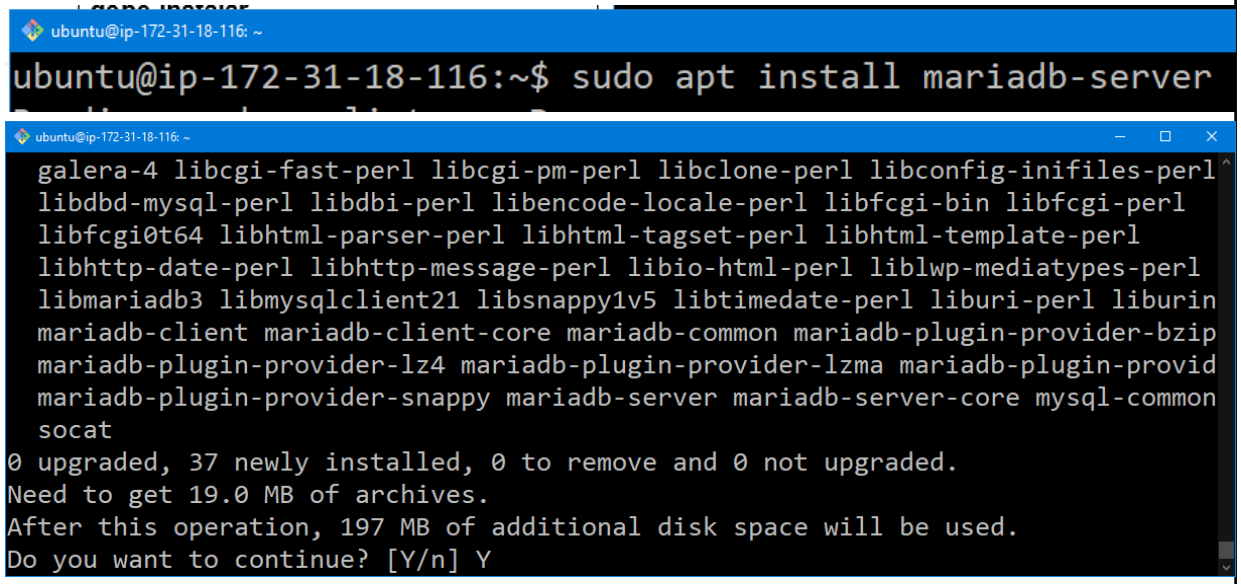
Verificar si existe el servicio de mysql
sudo service mysql status

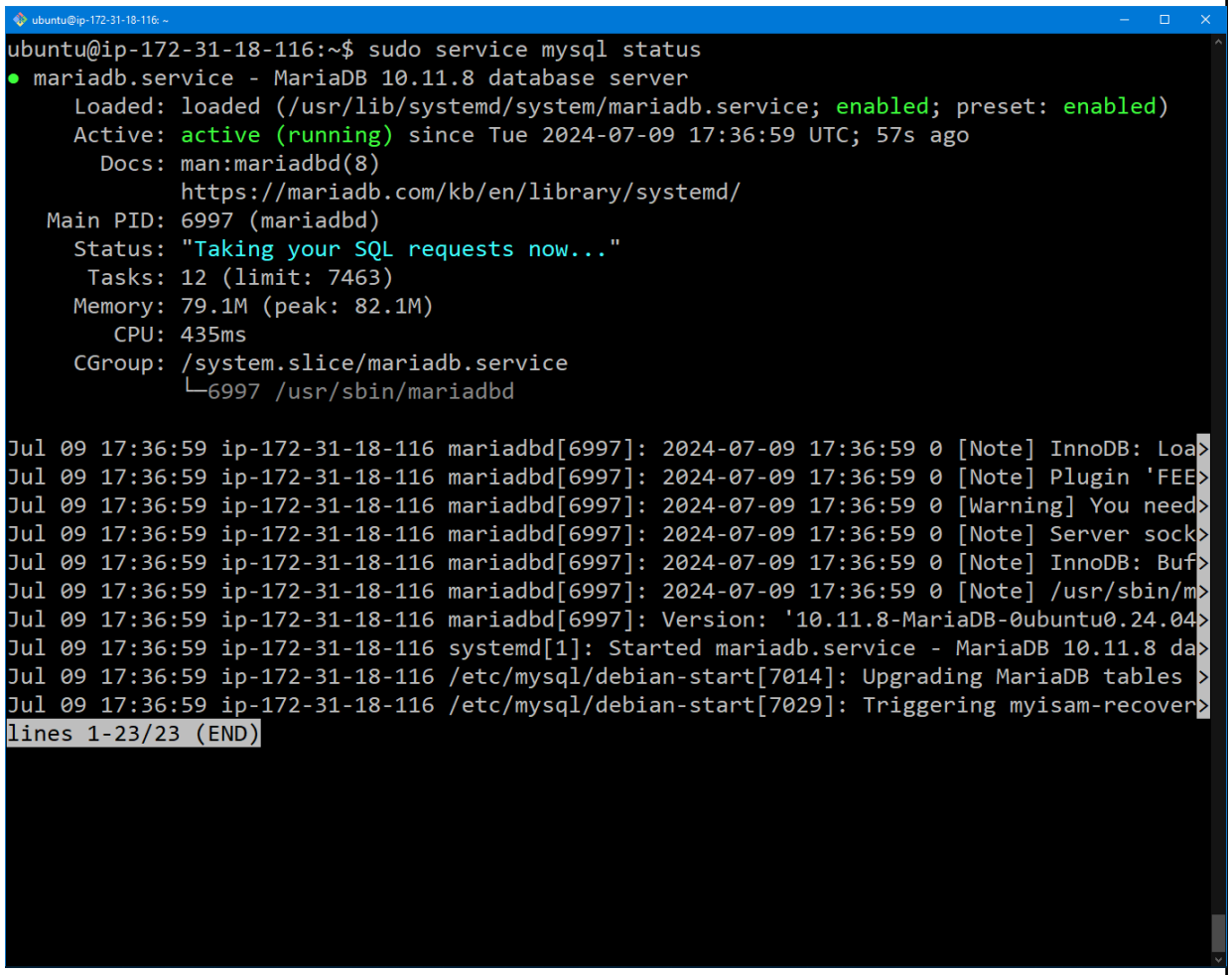
En caso de no encontrar el servicio se debe instalar

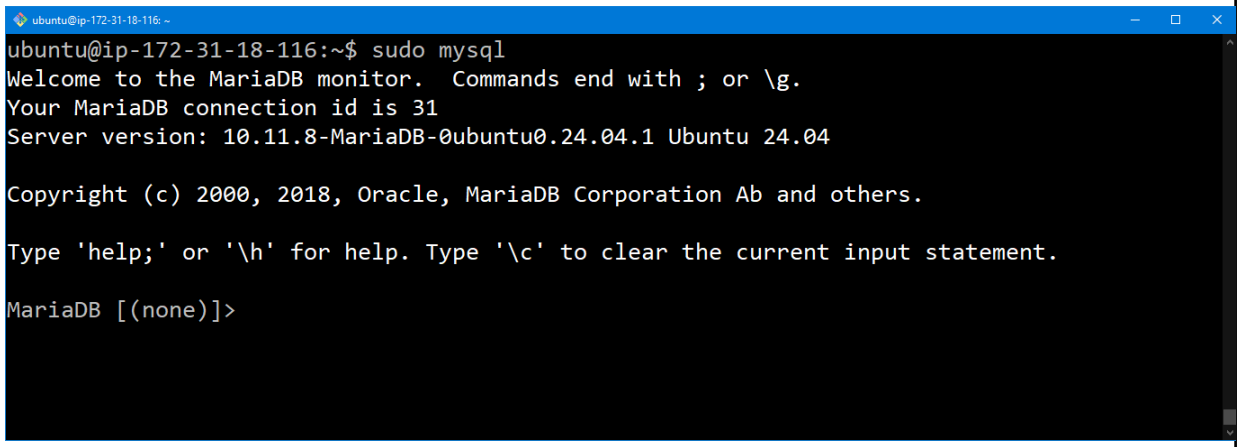
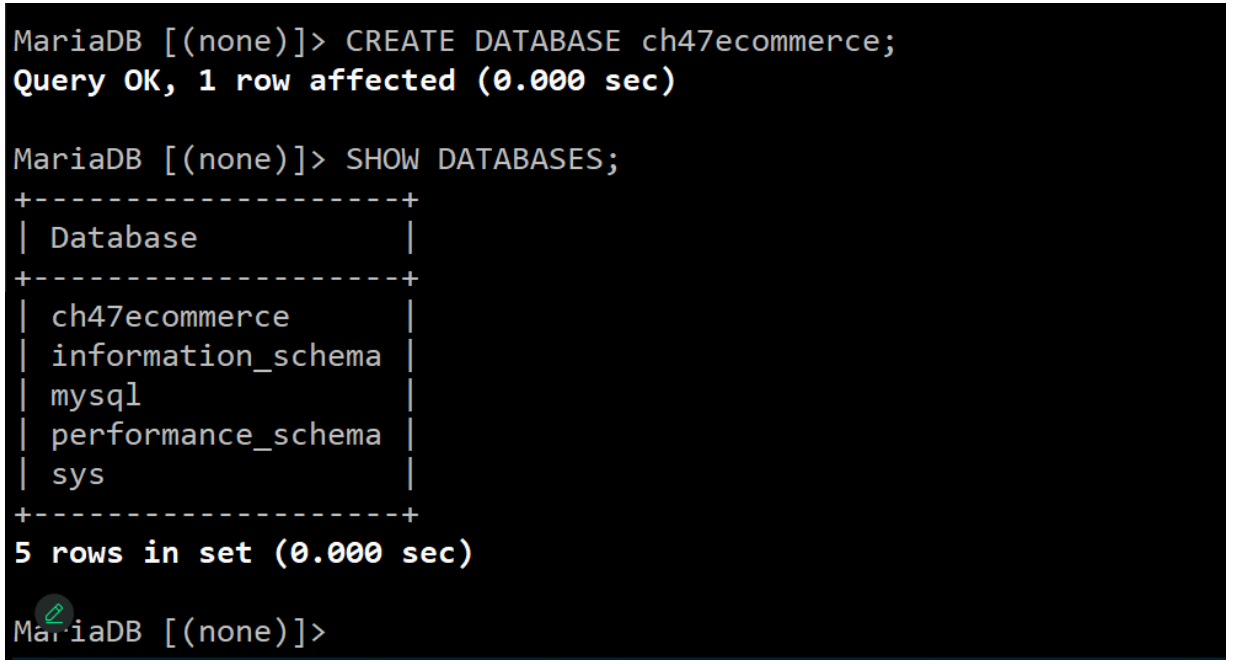


sudo apt install mariadb-server

En caso de preguntar por la instalación escribir **Y**



sudo service mysql status ***Nota: Si se abre un entorno como el del comando less, usar q para salir**	

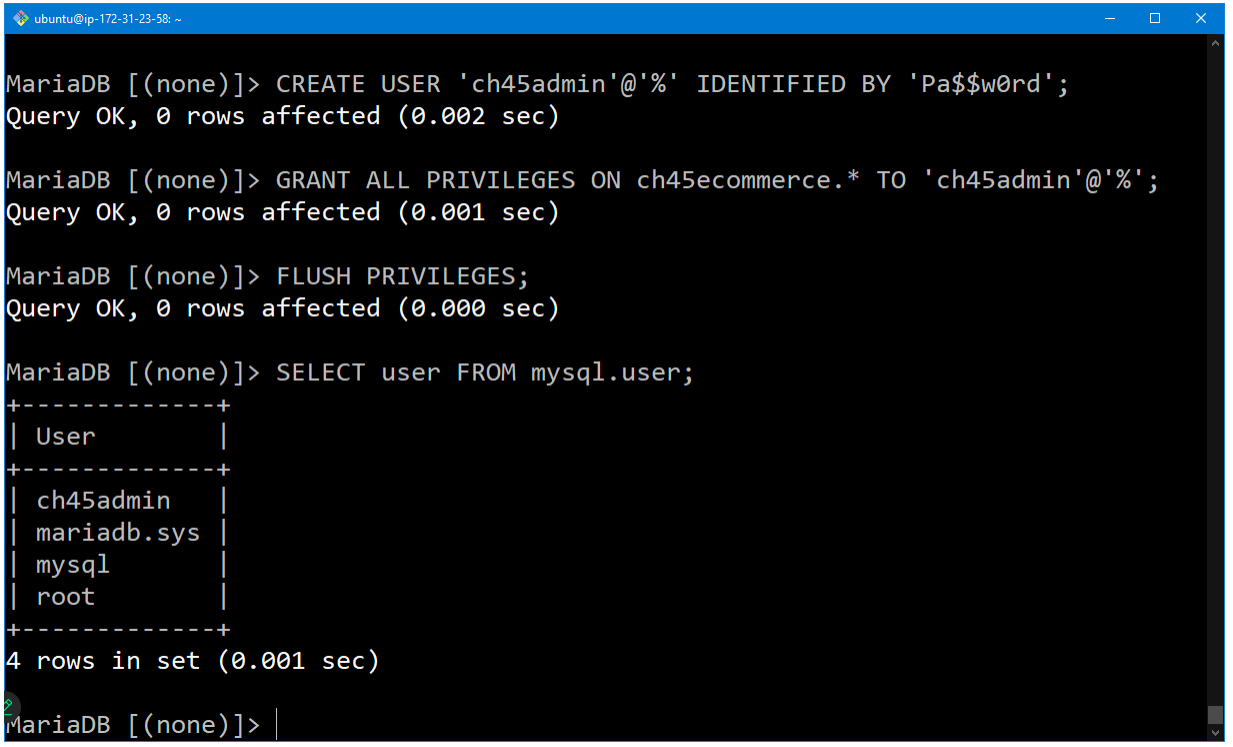
Para verificar la conexión utilizar sudo mysql	
Crear la base de datos ch47ecommerce **CREATE DATABASE ch47ecommerce;** Validar que la base de creó correctamente: **SHOW DATABASES;**	

Crear el usuario:
CREATE USER 'ch47admin'@'%' IDENTIFIED BY 'Pa$$w0rd';

Otorgar permisos al usuario sobre la base de datos:
GRANT ALL PRIVILEGES ON ch47ecommerce.* TO 'ch47admin'@'%';

Refrescar los permisos:
FLUSH PRIVILEGES;

Validar que el usuario se creó:
SELECT user FROM mysql.user;



Validar permisos del usuario creado en el paso anterior

SHOW GRANTS FOR 'ch47admin'@'%';

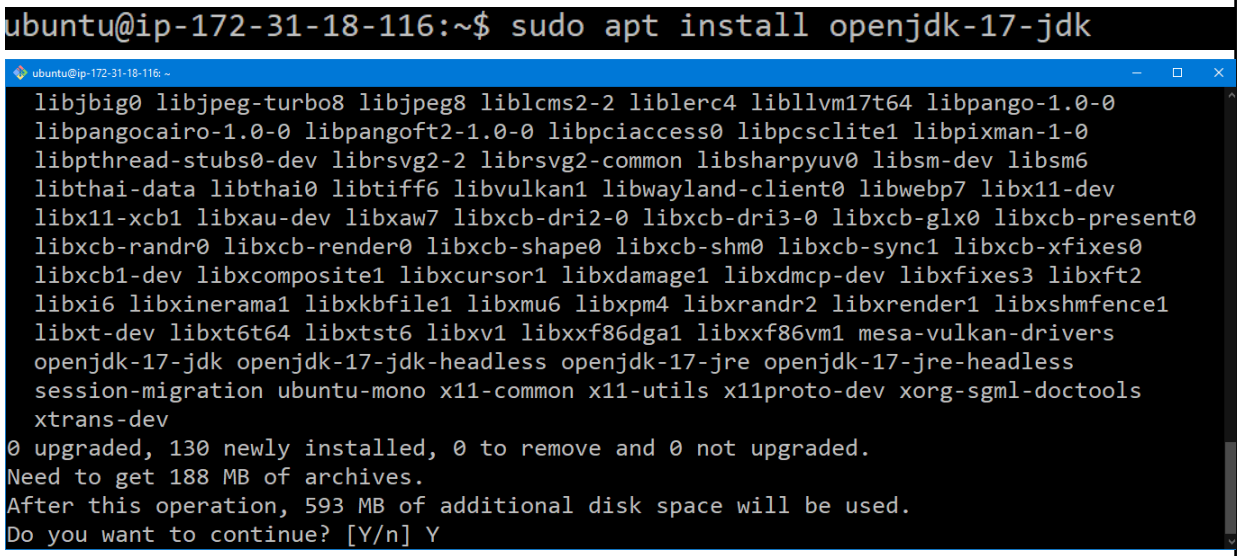
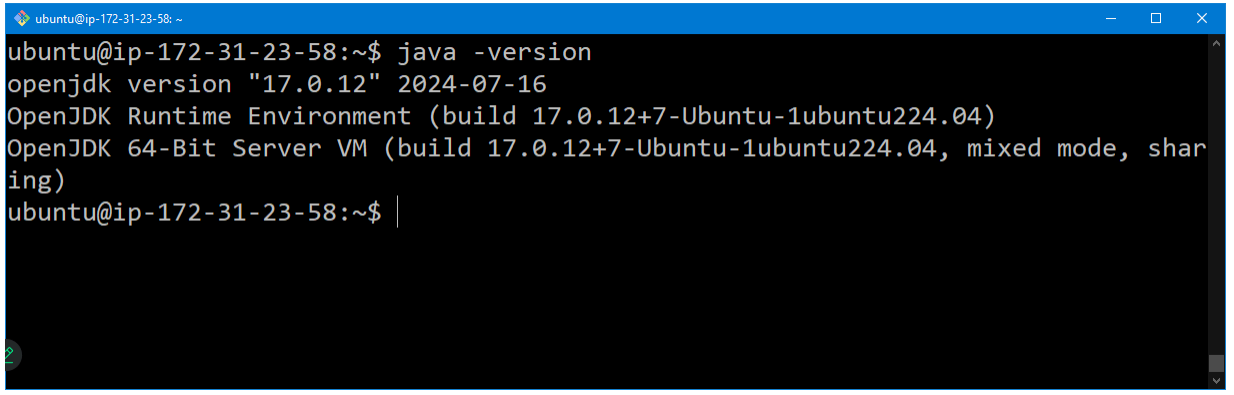
Usar el comando exit para salir de MySQL

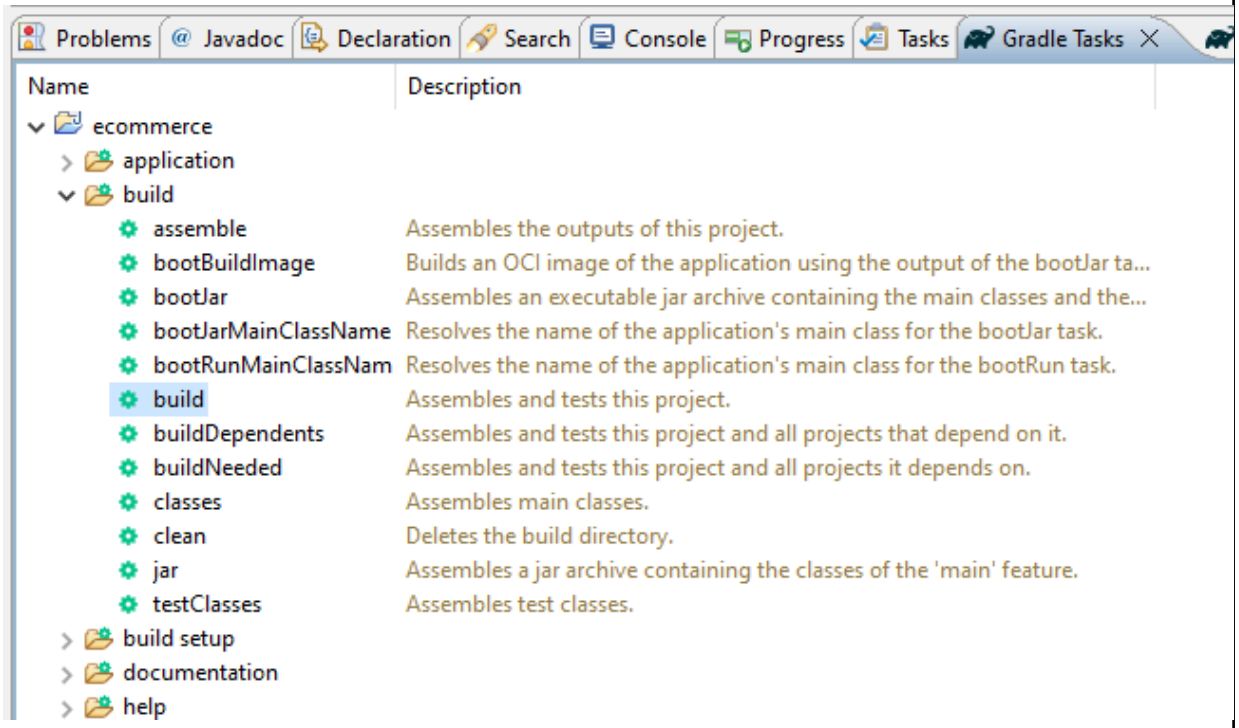
```
MariaDB [(none)]> SHOW GRANTS FOR 'ch47admin'@'%';
+-----------------------------------------------------------------------------------------------
-----------------+
| Grants for ch47admin@%
                 |
+-----------------------------------------------------------------------------------------------
-----------------+
| GRANT USAGE ON *.* TO `ch47admin`@`%` IDENTIFIED BY PASSWORD '*0F3704CF78312459ABAEBE40
557BFF7BFCB0A3EC' |
| GRANT ALL PRIVILEGES ON `ch47ecommerce`.* TO `ch47admin`@`%`
                 |
+-----------------------------------------------------------------------------------------------
-----------------+
2 rows in set (0.000 sec)

MariaDB [(none)]> exit
Bye
ubuntu@ip-172-31-13-120:~$
```

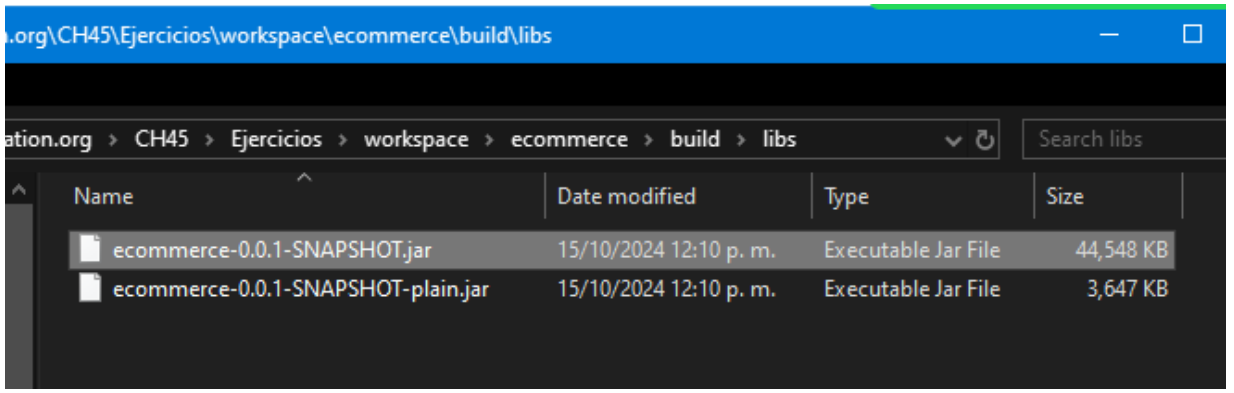
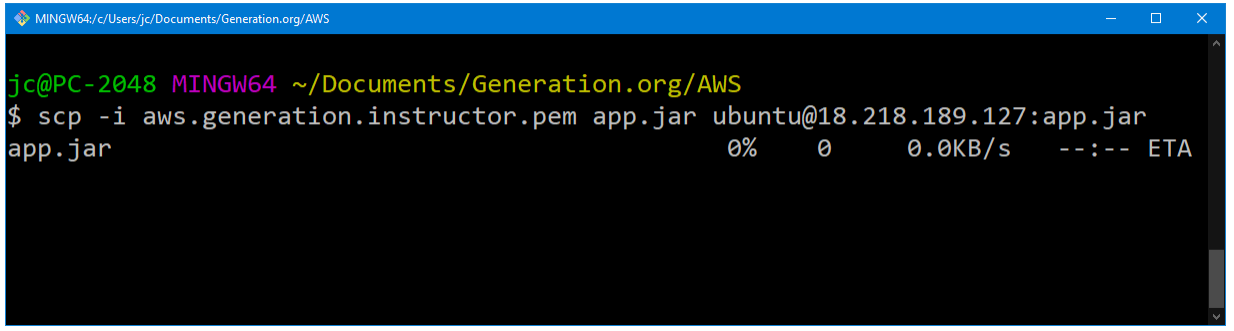
Instalación de java

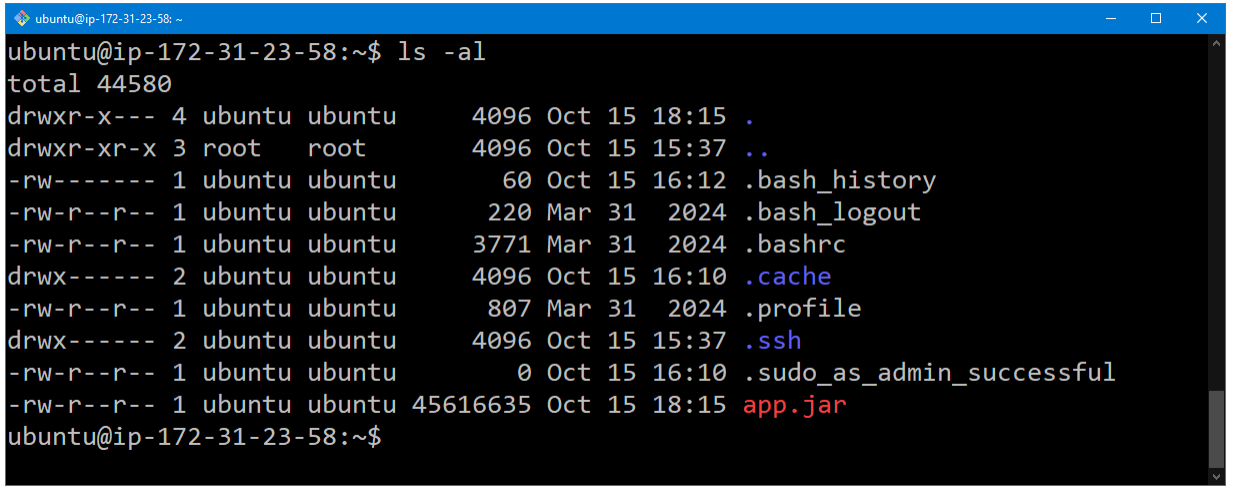
java -version

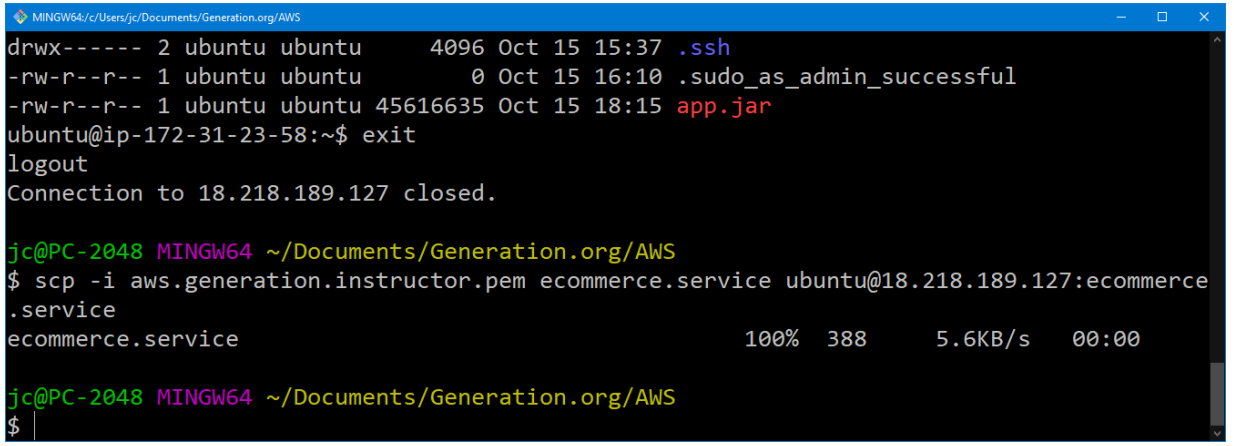
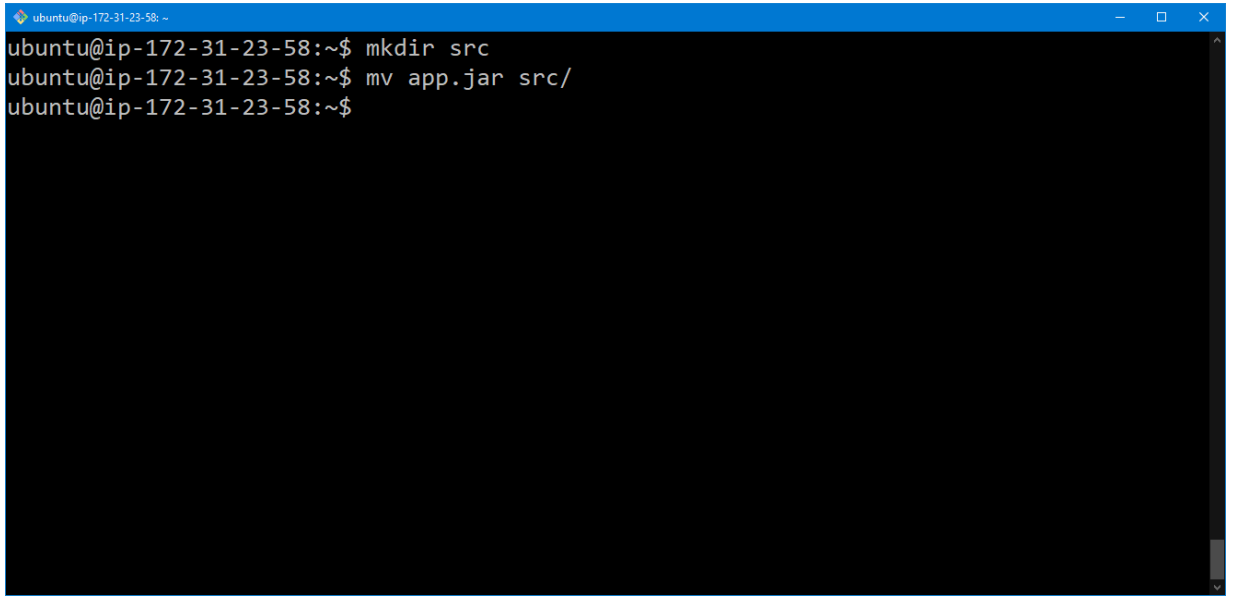
```
ubuntu@ip-172-31-18-116: ~
ubuntu@ip-172-31-18-116:~$ java -version
Command 'java' not found, but can be installed with:
sudo apt install default-jre              # version 2:1.17-75, or
sudo apt install openjdk-17-jre-headless  # version 17.0.10~6ea-1
sudo apt install openjdk-8-jre-headless   # version 8u412-ga-1~24.04.2
sudo apt install openjdk-11-jre-headless  # version 11.0.21+9-0ubuntu1
sudo apt install openjdk-19-jre-headless  # version 19.0.2+7-4
sudo apt install openjdk-20-jre-headless  # version 20.0.2+9-1
sudo apt install openjdk-21-jre-headless  # version 21.0.1+12-3
sudo apt install openjdk-22-jre-headless  # version 22~22ea-1
ubuntu@ip-172-31-18-116:~$
```

En el caso de no tener instalado java, ejecutar la instalación: **sudo apt install openjdk-17-jdk** En caso de preguntar por la instalación escribir **Y**	
Validar que se instaló correctamente con: **java -version**	

Modificar el application.properties para que tenga variables de entorno en lugar de valores	``` spring.application.name=ecommerce #Información para conectarse al servidor spring.datasource.url=jdbc:mysql://${PROD_DB_HOST}:3306/${PROD_DB_NAME} spring.datasource.username=${PROD_DB_USERNAME} spring.datasource.password=${PROD_DB_PASSWORD} #create, create-drop, validate, update spring.jpa.hibernate.ddl-auto=${PROD_DDL} ```
Subir el archivo jar de la aplicación ecommerce (build/libs/ecommerce-0.0.1-SNAPSHOT.jar) Generar el jar con doble click en: **Gradle Task-> ecommerce -> build -> build**	

Buscar y renombra el archivo por app.jar	
Salir de la conexión del servidor con: **exit** Copiar a la ruta de las llaves el archivo app.jar Subir el archivo al servidor: **scp -i archivollave.pem app.jar ubuntu@direccionIP:app.jar** Volverse a conectar y validar que el archivo existe con un **ls -al**	

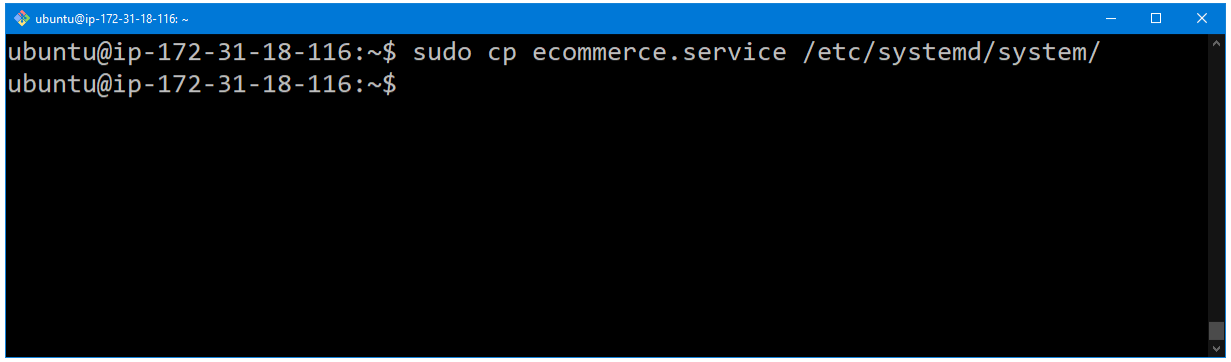
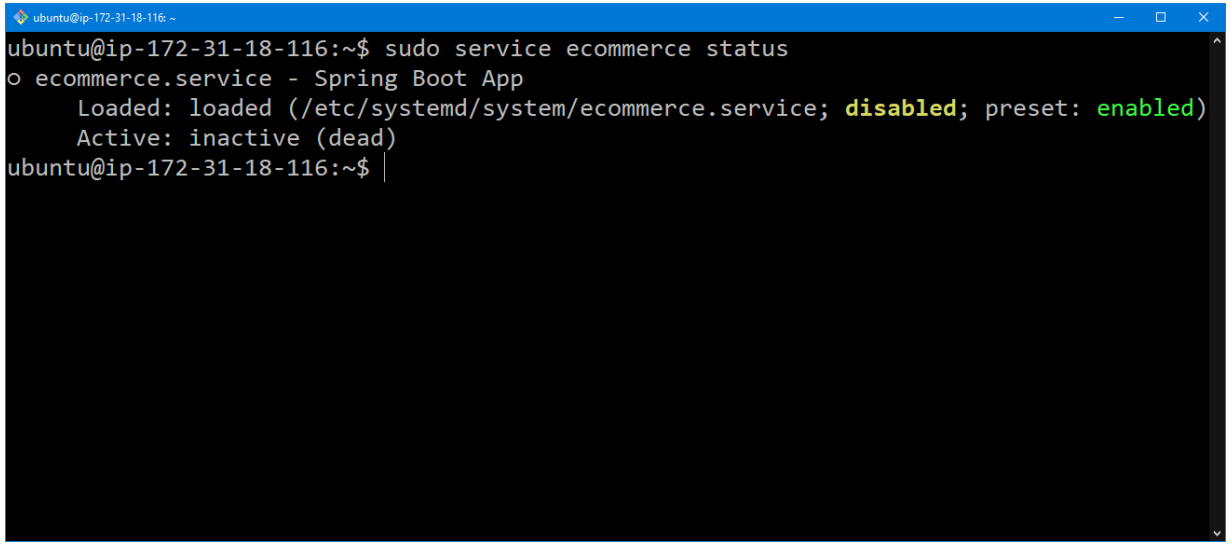
	
archivo ecommerce.service	[Unit] Description=Spring Boot App After=syslog.target [Service] User=ubuntu ExecStart=java -jar /home/ubuntu/src/app.jar Environment="PROD_DB_HOST=localhost" Environment="PROD_DB_NAME=ch47ecommerce" Environment="PROD_DB_USERNAME=ch47admin" Environment="PROD_DB_PASSWORD=Pa$$w0rd" Environment="PROD_DDL=update" SuccessExitStatus=143 [Install] WantedBy=multi-user.target

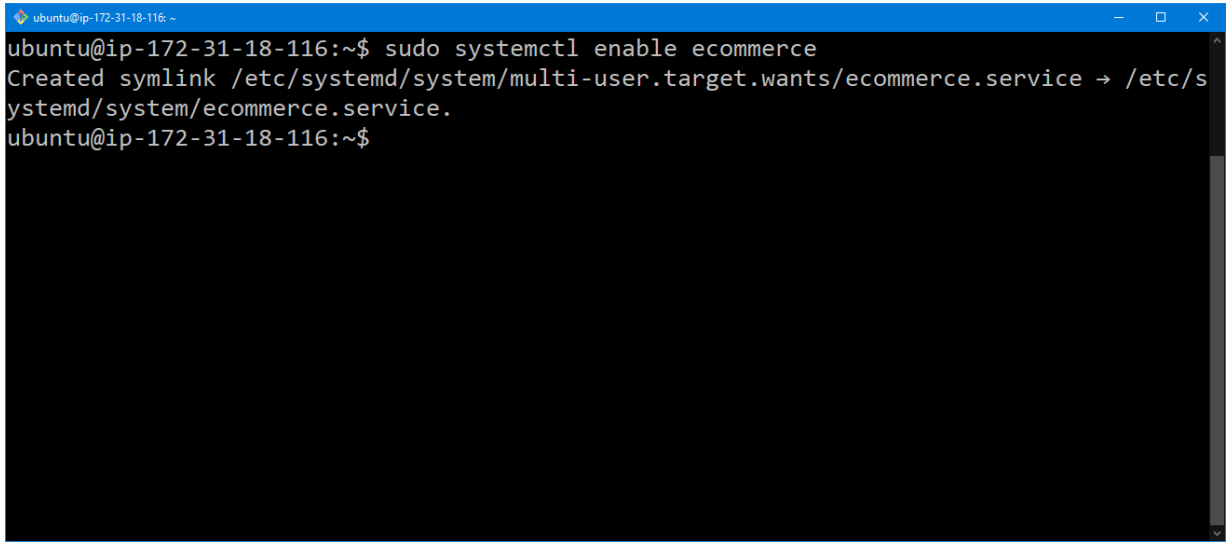
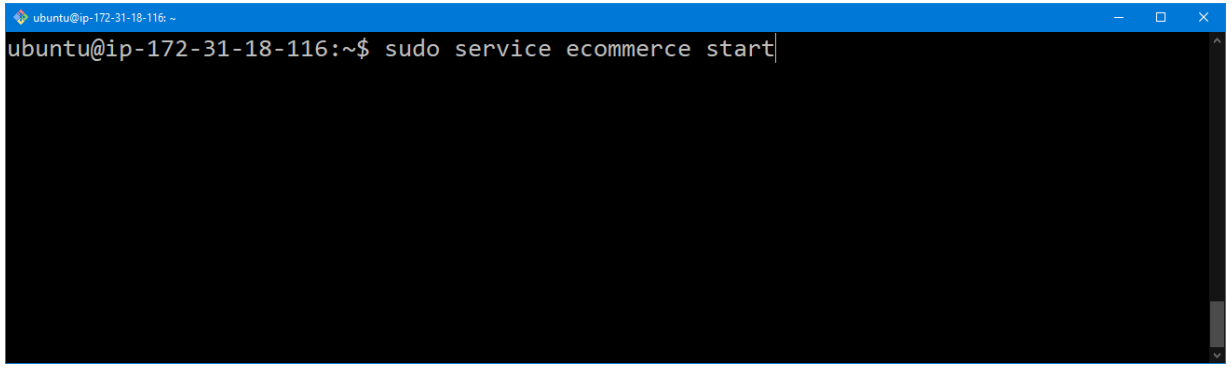
Subir el archivo ecommerce.service con el comando **scp -i archivollave.pem ecommerce.service ubuntu@direccionIP:ecommerce.service**	
Volver a conectarse al servidor (como en el paso 1) Mover el archivo app.jar al directorio src	

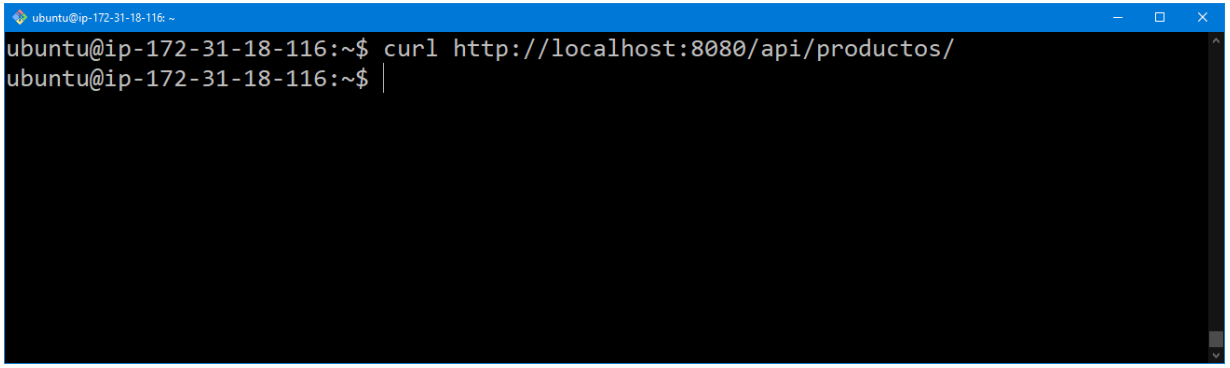
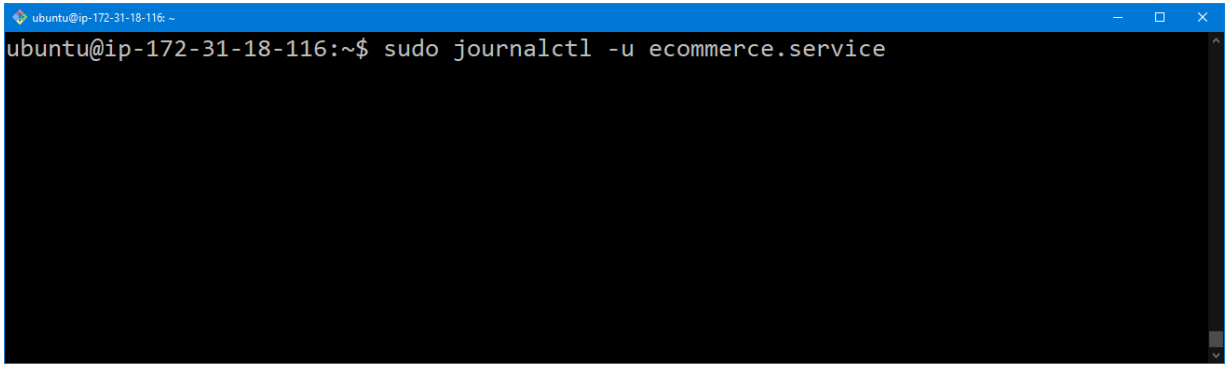
Volver a conectarse al servidor y validar que el archivo existe.

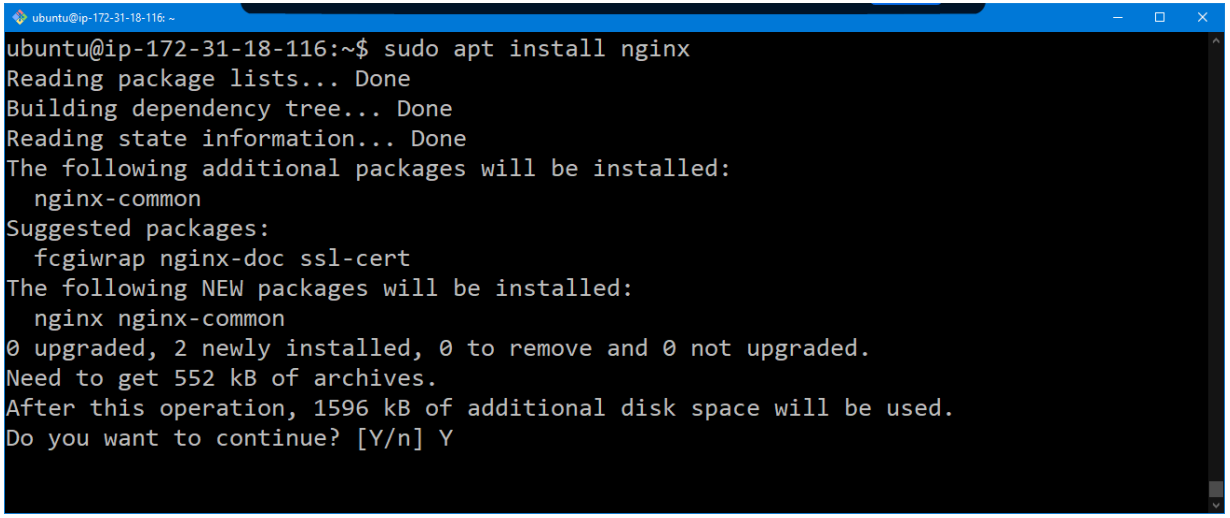
cat ecommerce.service

```
ubuntu@ip-172-31-13-120:~$ cat ecommerce.service
[Unit]
Description=Spring Boot App
After=syslog.target

[Service]
User=ubuntu
ExecStart=java -jar /home/ubuntu/src/app.jar
Environment="PROD_DB_HOST=localhost"
Environment="PROD_DB_NAME=ch47ecommerce"
Environment="PROD_DB_USERNAME=ch47admin"
Environment="PROD_DB_PASSWORD=Pa$$w0rd"
Environment="PROD_DDL=update"

SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
```

Copiar el ecommerce.service en los servicios del servidor. **sudo cp ecommerce.service /etc/systemd/system/**	
Validar el estado del servicio se utiliza: **sudo service ecommerce status** Si muestra una pantalla completa con información, nos podemos salir presionando la tecla **q**	

Para habilitar el servicio y que inicie cuando el sistema operativo inicia, se utiliza el comando: **sudo systemctl enable ecommerce**	
Iniciar el servicio: **sudo service ecommerce start**	

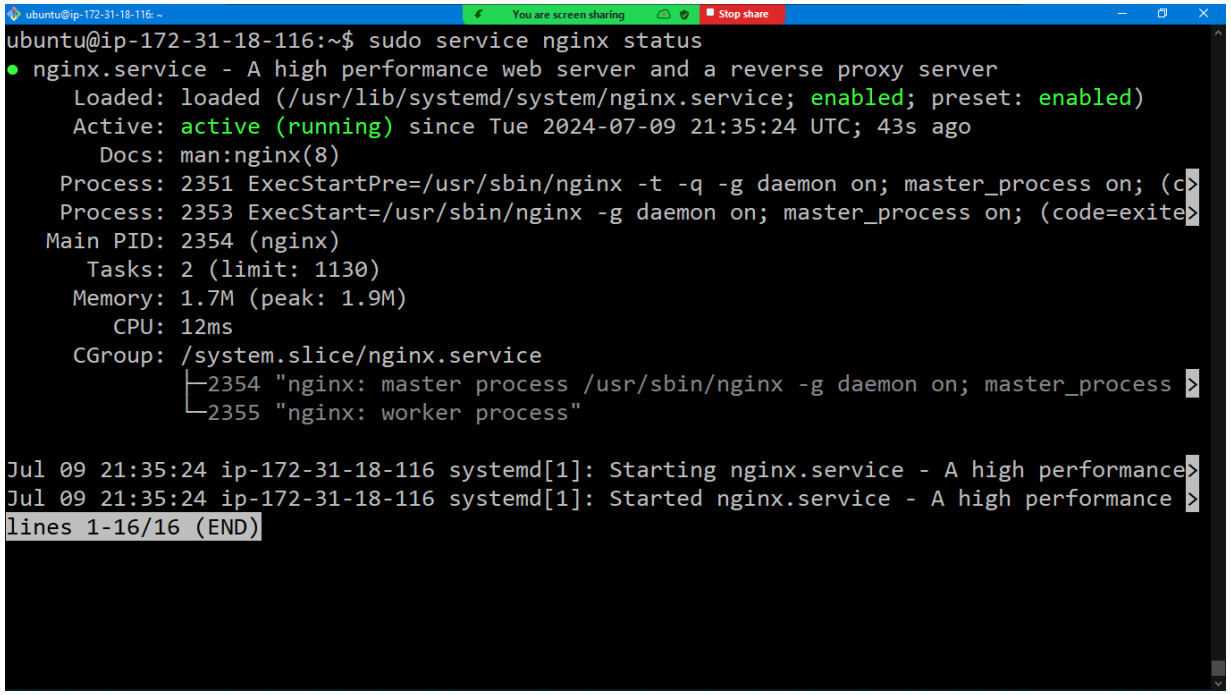
Validar que el servicio responde: **curl http://localhost:8080/api/productos/**	
Si el servicio no responde, se puede revisar el log del servicio con el comando: **sudo journalctl -u ecommerce.service** Si muestra mucho contenido pueden usar la tecla q para salir. Tecla G para ir al final Tecla g para ir al principio	

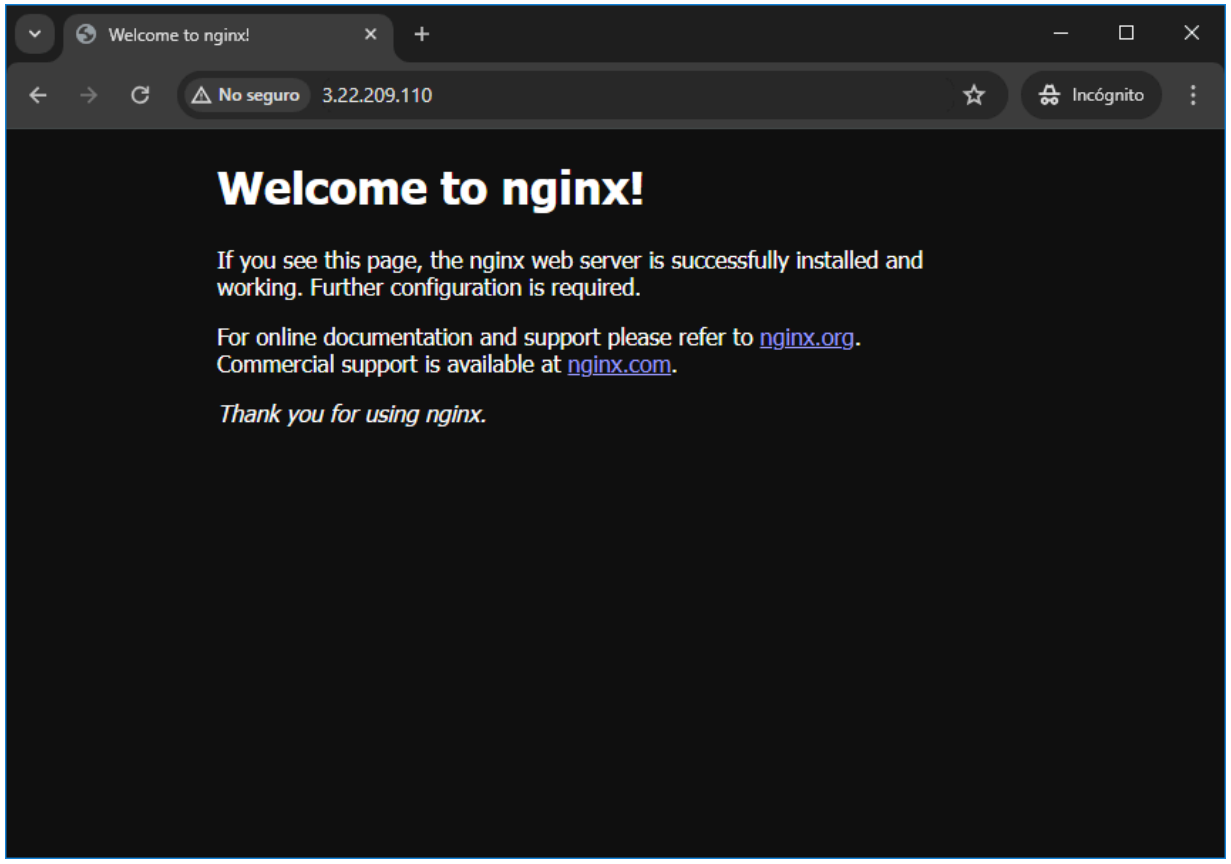
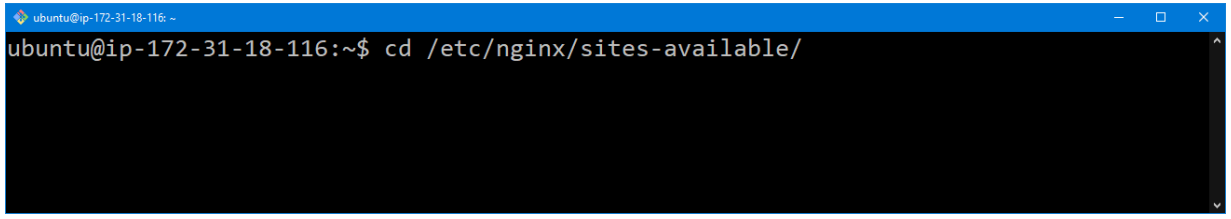
Instalar el servidor nginx (proxy) para mapear el puerto 80 al puerto 8080 **sudo apt install nginx** Presionar la **Y** para continuar con la instalación	

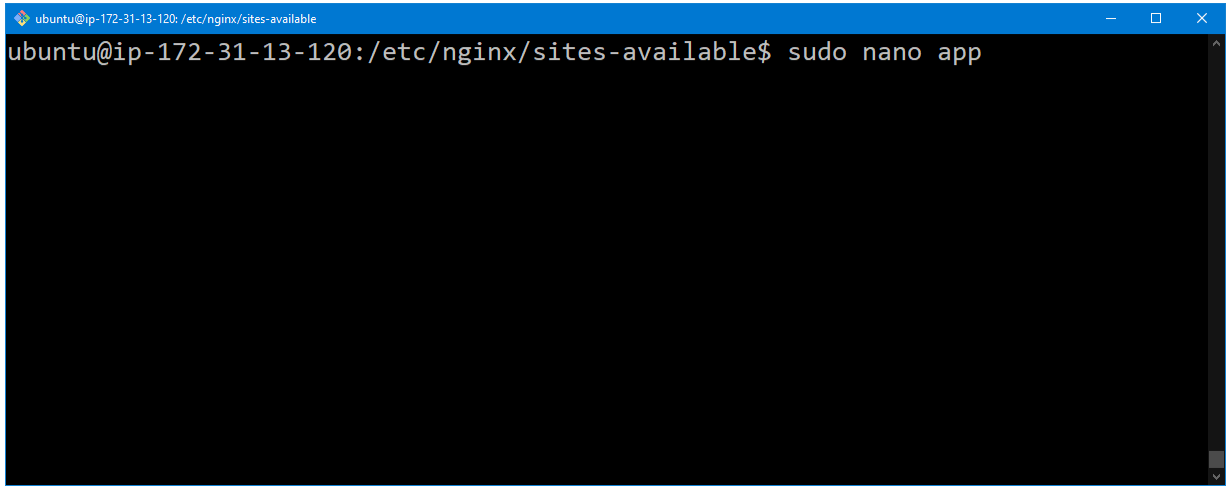
Validar que el servicio nginx está levantado:

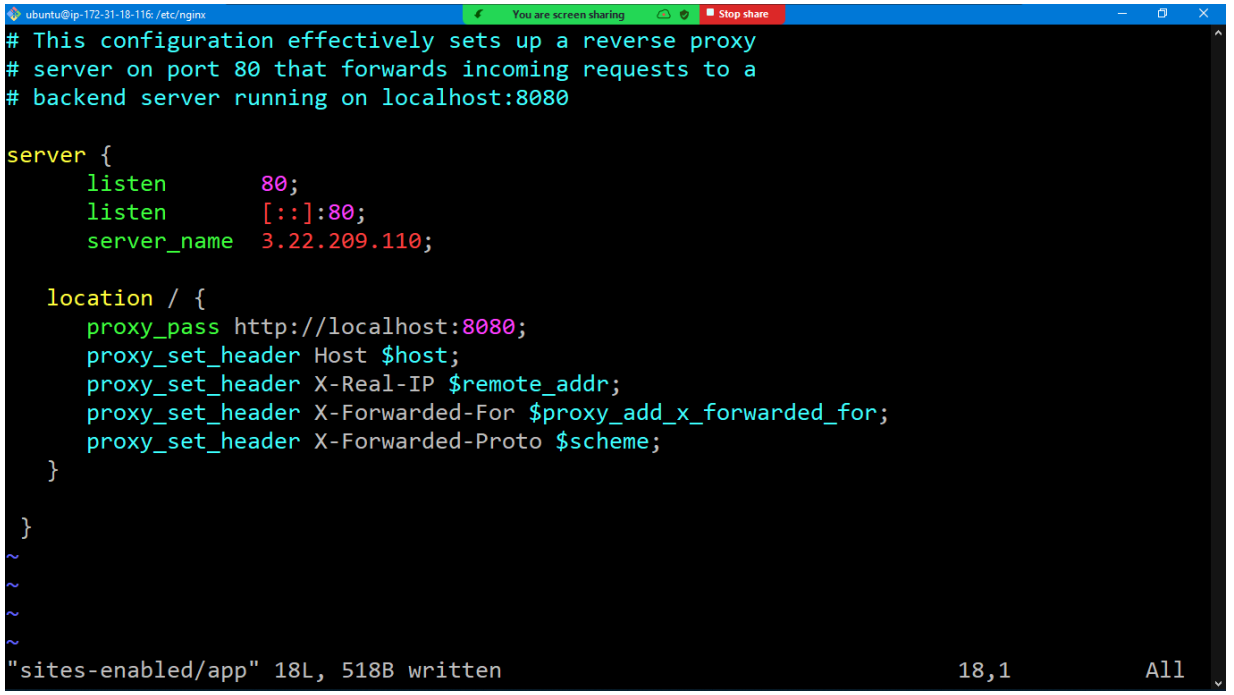
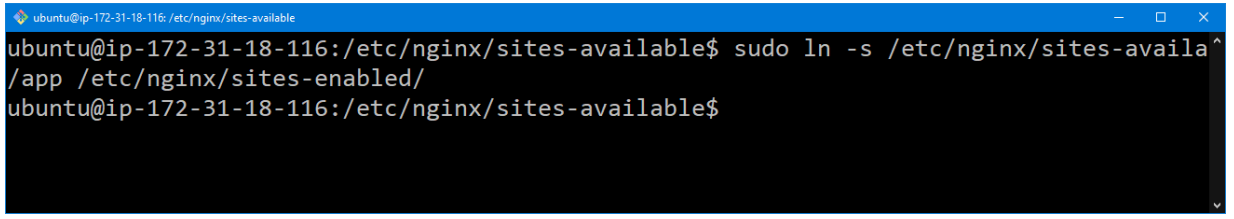
sudo service nginx status

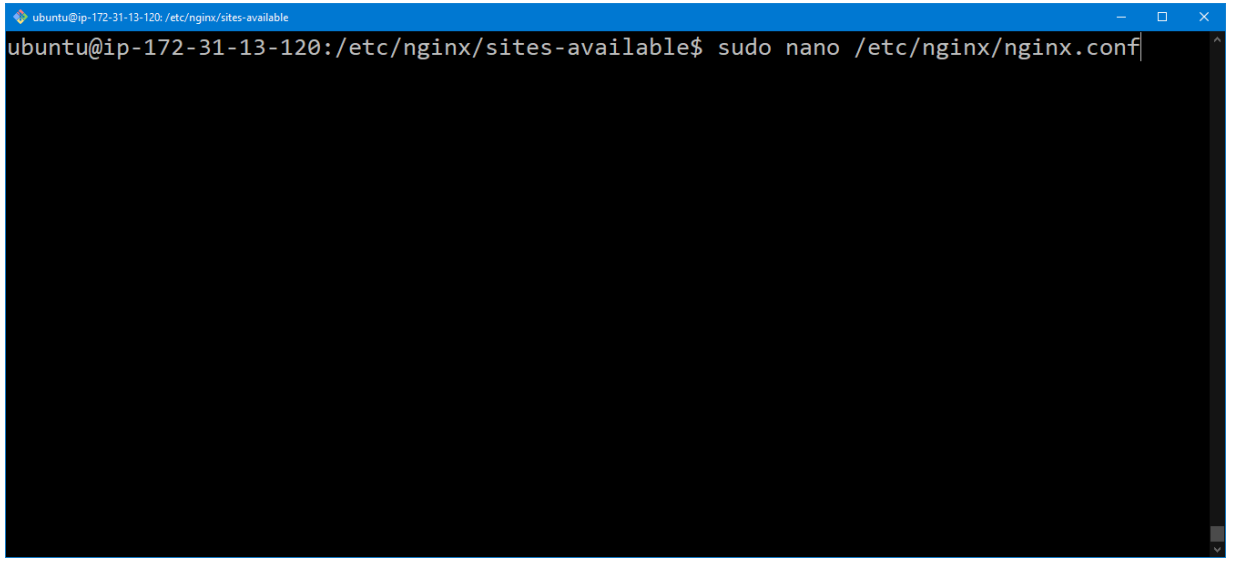
También se puede validar con la dirección ip en el navegador y veremos la página default de nginx.

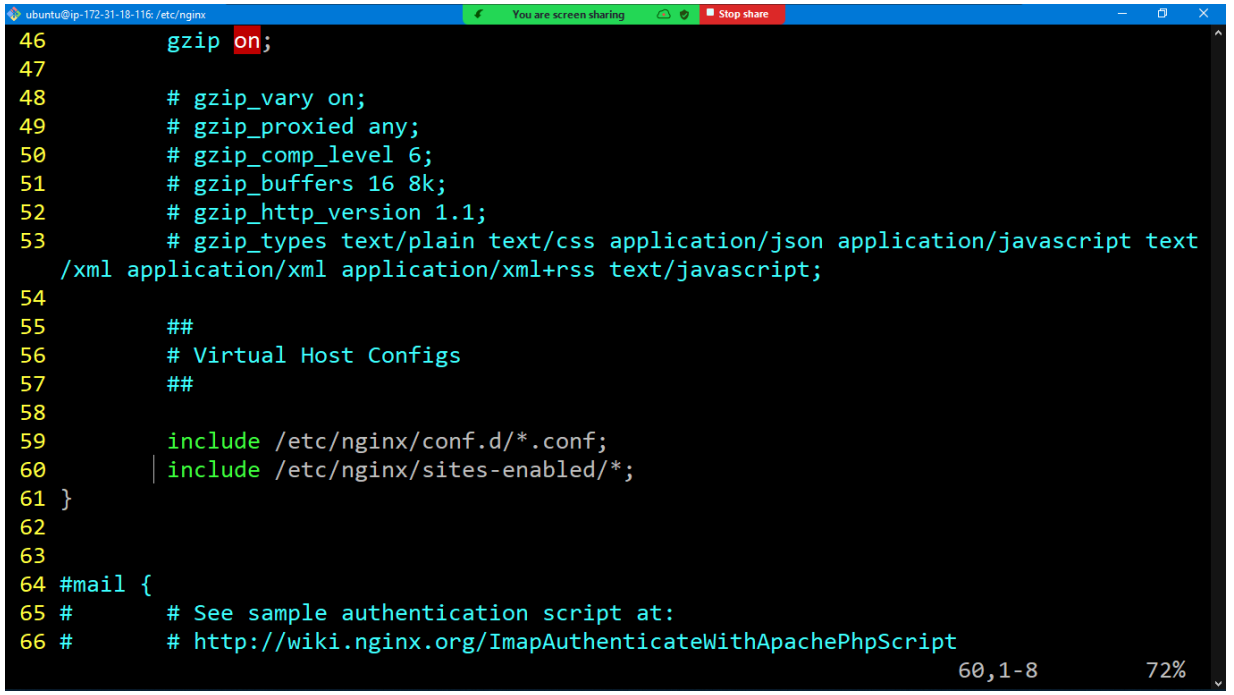
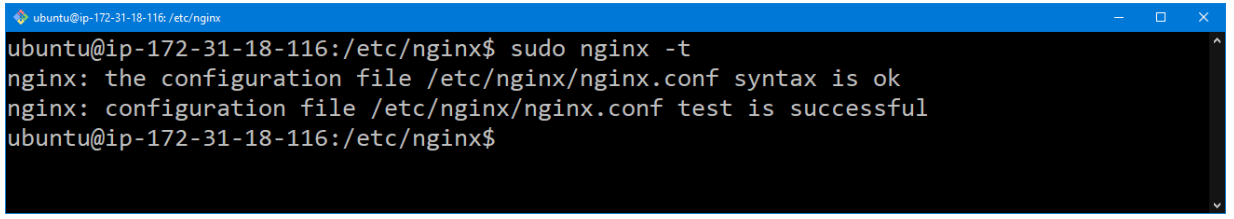


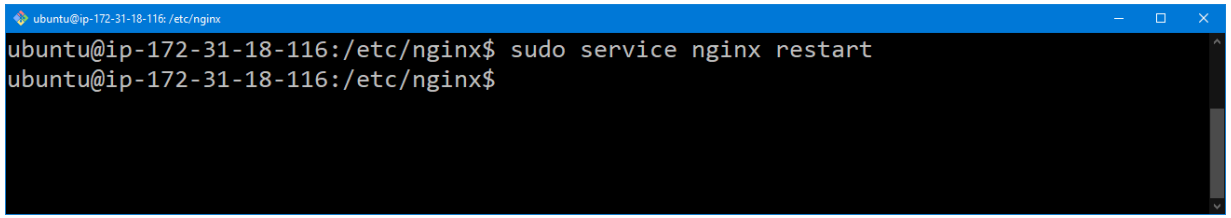
	
Cambiarse al directorio de configuración de nginx: **cd /etc/nginx/sites-available/**	

Crear archivo de configuración: **sudo nano app**	
Pegar la siguiente información Cambiar el **localhost** de server_name por la dirección ip del servidor	# This configuration effectively sets up a reverse proxy # server on port 80 that forwards incoming requests to a # backend server running on localhost:8080 server { listen 80; listen [::]:80; server_name **localhost**; location / { proxy_pass http://localhost:8080; proxy_set_header Host $host; proxy_set_header X-Real-IP $remote_addr; proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; proxy_set_header X-Forwarded-Proto $scheme; } }

	
Crear el enlace en los sitios habilitados: **sudo ln -s /etc/nginx/sites-available/app /etc/nginx/sites-enabled/**	

Cambiar la configuración principal de nginx: **sudo nano /etc/nginx/nginx.conf**	

En la sección http **verificar** que se encuentre la línea 60, si no se encuentra se debe agregar después de la línea: **include /etc/nginx/conf.d/*.conf;** **include /etc/nginx/sites-enabled/*;**	
Para validar la sintaxis de las configuraciones de nginx utilizamos: **sudo nginx -t**	

Reiniciar servicio de nginx: **sudo service nginx restart**	
Probar la aplicación En caso de mostrar mensaje de sitio inseguro darle en Continuar al sitio.	http://ip