31-1-2021

PATRICIA VILLANUEVA POLO

Actividad 2 – Despliegue aplicaciones web

Índice

[I. Instalación de Java en el servidor Ubuntu 2](#_Toc63017224)

[II. Instalar OpenSSH en servidor Ubuntu 5](#_Toc63017225)

[III. Instalación de Apache en el servidor Ubuntu 6](#_Toc63017226)

[1. Requisitos previos 6](#_Toc63017227)

[Crear un usuario nuevo 6](#_Toc63017228)

[Configuración Firewall 6](#_Toc63017229)

[2. Instalando Apache 7](#_Toc63017230)

[IV. Instalar Tomcat en servidor Ubuntu 9](#_Toc63017231)

[V. Instalar MariaDB en servidor Ubuntu 10](#_Toc63017232)

[VI. Comprobación, configuración y accesibilidad 13](#_Toc63017233)

[1. Apache 15](#_Toc63017234)

[2. Tomcat 17](#_Toc63017235)

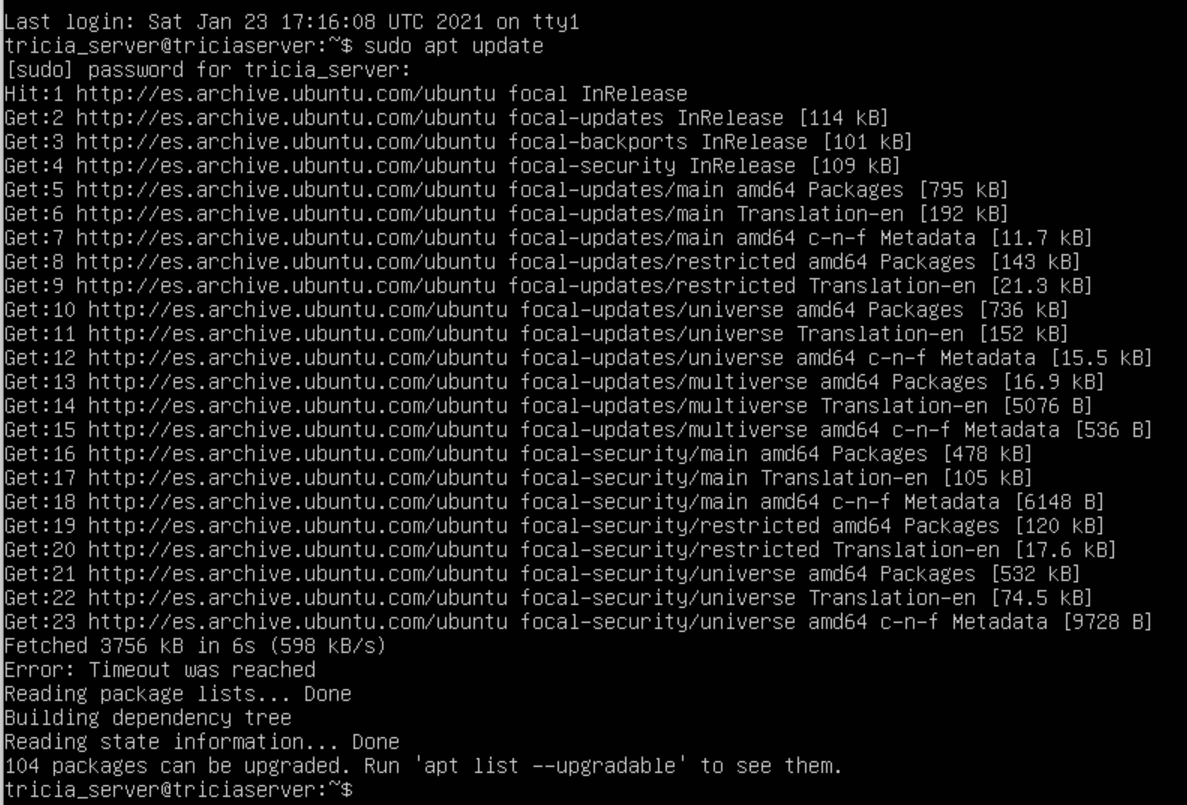
[3. Mariadb 18](#_Toc63017236)

# Instalación de Java en el servidor Ubuntu

Por defecto, Ubuntu 20.04 incluye Open JDK 11, que es una variante de código abierto de JRE y JDK.

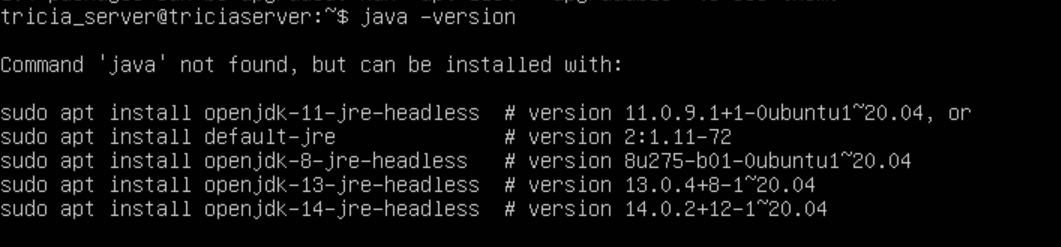
Para instalar esta versión primero actualizaremos el índice de paquetes:

Sudo apt update



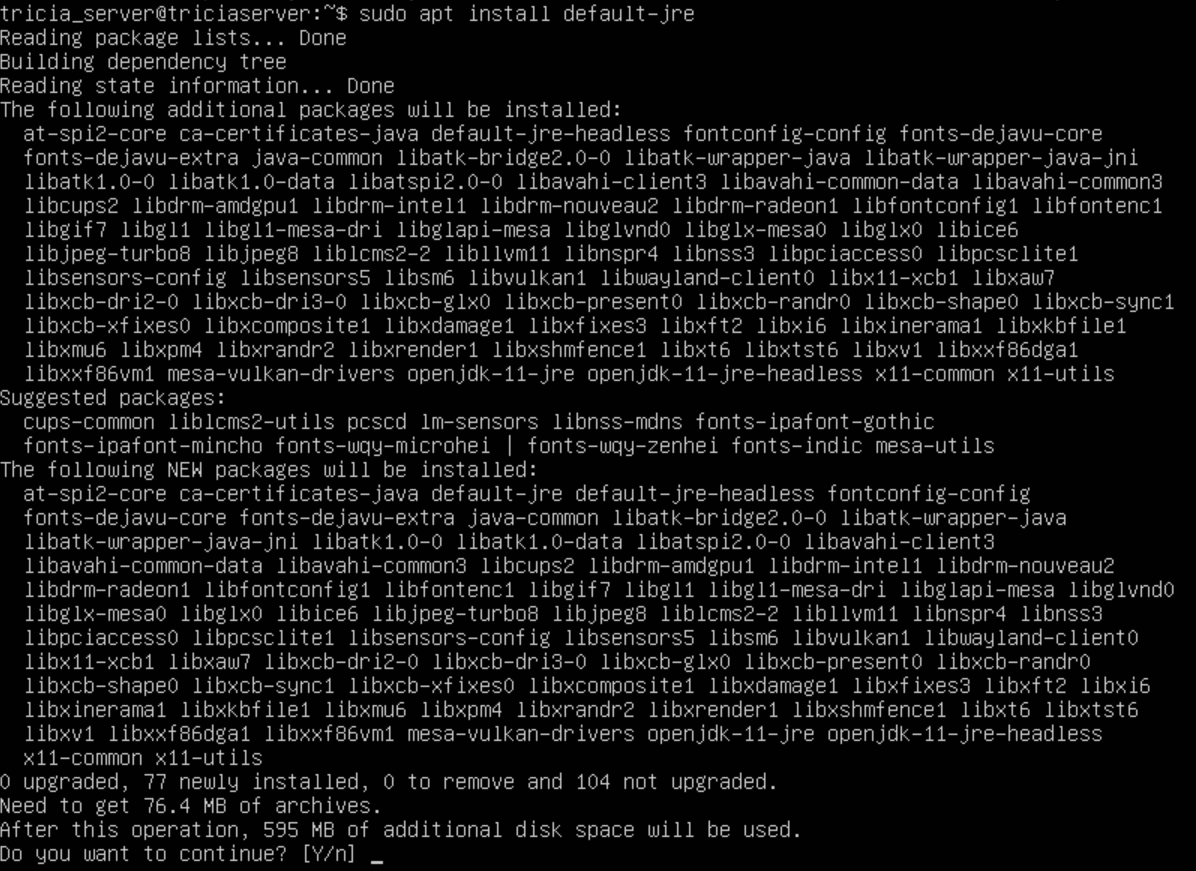
A continuación, nos aseguramos de si tenemos Java instalado:

Java -version

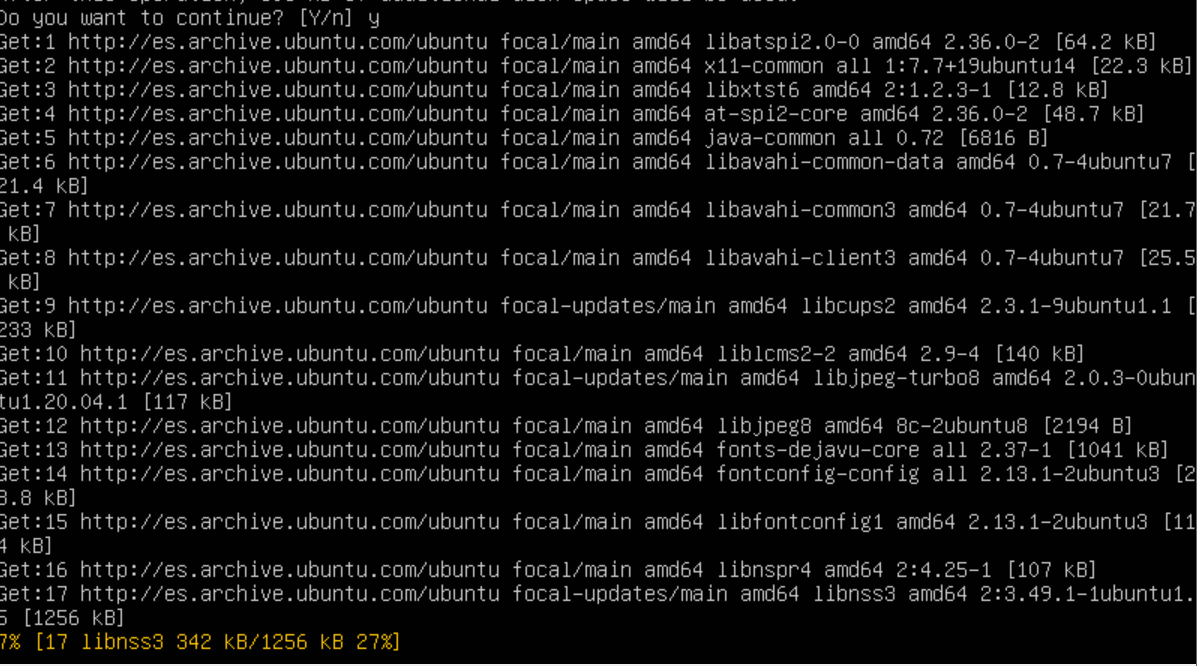


Nos indica que no lo está, así que vamos a pasar a instalarlo. Empezamos por el JRE:

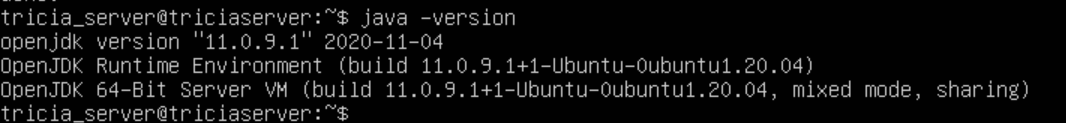
sudo apt install default-jre



Le damos a Y + Enter y empieza a instalar

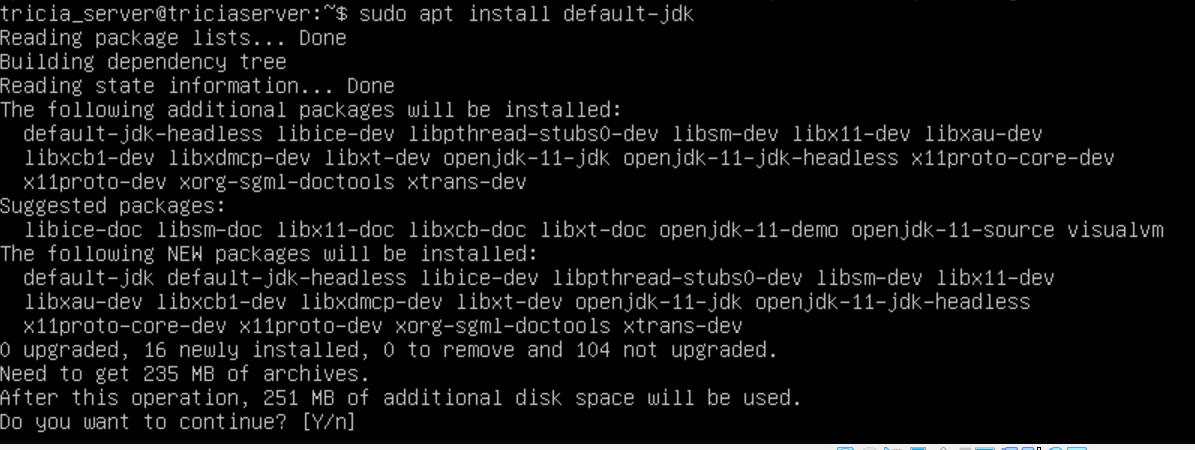


Si volvemos a comprobar con **java -version** veremos lo siguiente:

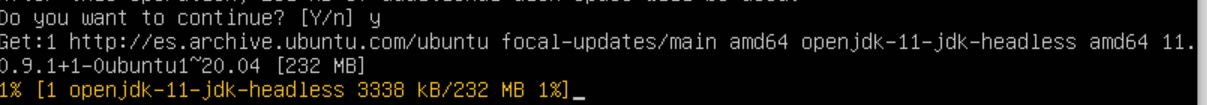


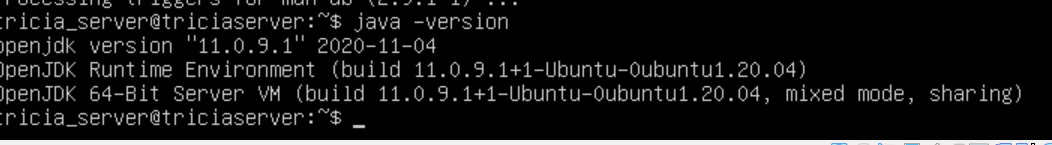
Ahora vamos a instalar también el JDK

Sudo apt install default-jdk

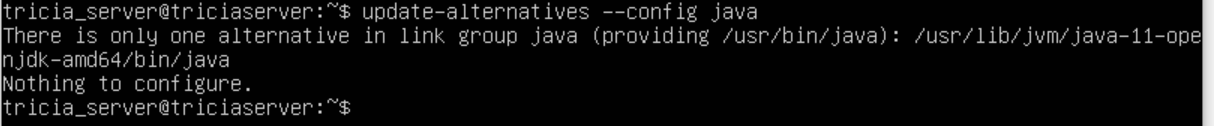


Damos a Y y enter para que comience la instalación:





Lo siguiente será generar la variable de entorno JAVA\_HOME que muchas aplicaciones utilizan. Si tecleamos echo JAVA\_HOME y no aparece nada, es porque no tenemos creado la variable de entorno. Debería aparecer una url. Para solucionarlo sacamos la ruta en la que hemos instalado java:



Copiamos la ruta: /usr/lib/jvm/java-11-openjdk-amd64/ (sin bin ni lo que vaya a continuación)

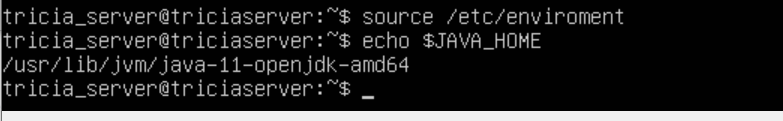
Abrimos /etc/enviroment con nano (editor) y tecleamos lo siguiente:

**JAVA\_HOME=”/usr/lib/jvm/java-11-openjdk-amd64”**

Pulsamos Ctrl+x para que se quede guardado el cambio y salir de nano. Ahora recargamos el archivo para que tenga efecto en nuestra sesión:

source /etc/enviroment



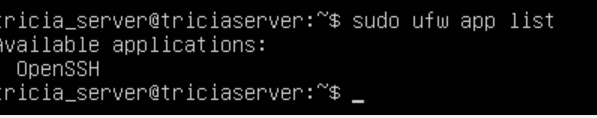


Con **echo $JAVA\_HOME** verificamos que ahora esté cargada la variable de entorno.

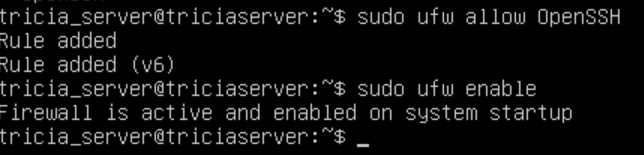
# Instalar OpenSSH en servidor Ubuntu

En mi caso ya lo tenía instalado de la Actividad 1. Para confirmarlo vamos a ver si tenemos el servicio.

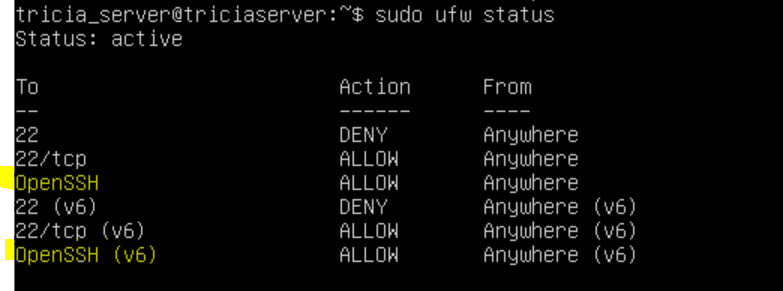
Sudo ufw app list



Me muestra que tengo instalado OpenSSH, para instalarlo lo hice con **sudo apt install -y ssh** y el comando **systemctl status ssh** me permite ver si está o no activo. Para asegurarnos de que va a estar funcionando también puedo usar el comando: **sudo ufw allow OpenSSH**. Y por último lo habilitamos con **sudo ufw enable**.



Sudo ufw status vemos si ya lo tenemos activado y permite la conexión:



# Instalación de Apache en el servidor Ubuntu

## Requisitos previos

Para instalar Apache debemos tener un usuario que no sea root, pero tenga permisos de administrador y también un firewall configurado con los puertos no esenciales cerrados.

### Crear un usuario nuevo

Nosotros ya tenemos un usuario así: tricia\_server y que hemos creado en la actividad 1. Pero para crear uno nuevo solo haría falta lo siguiente:

Sudo adduser nomUsuario

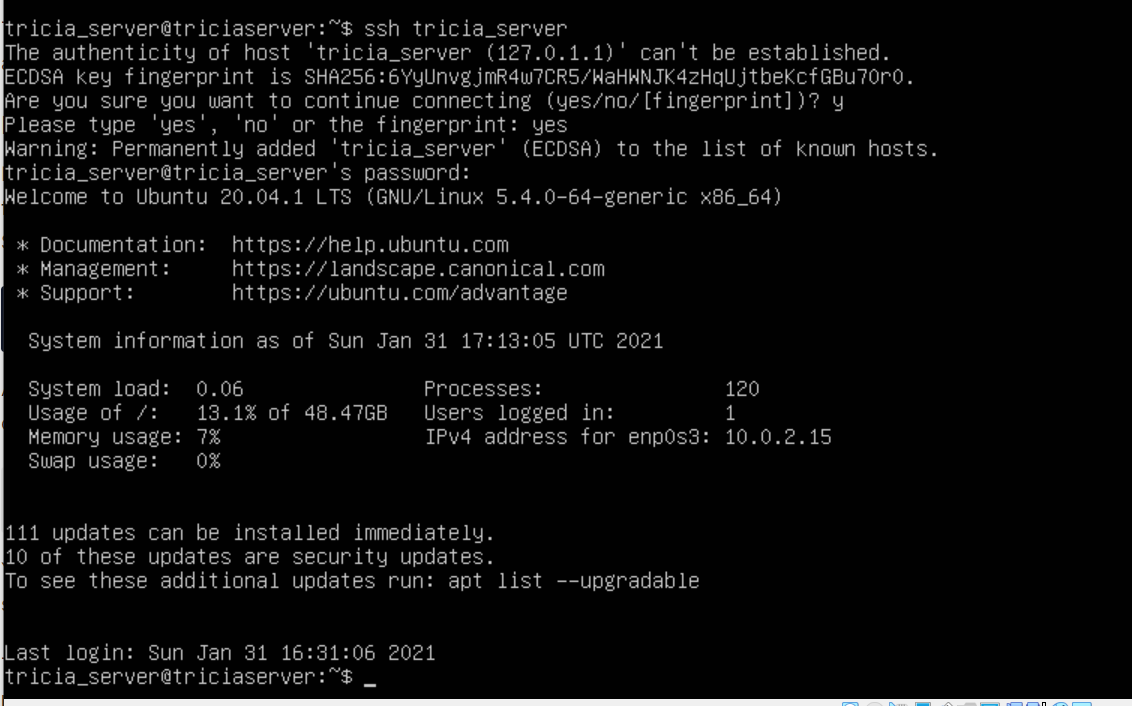
Luego debemos meterlo en el grupo sudo para que tenga permisos de adminsitrador:

usermod -aG sudo user\_apch

### Configuración Firewall

Nos aseguramos de tener acceso al servicio ssh desde nuestro usuario:

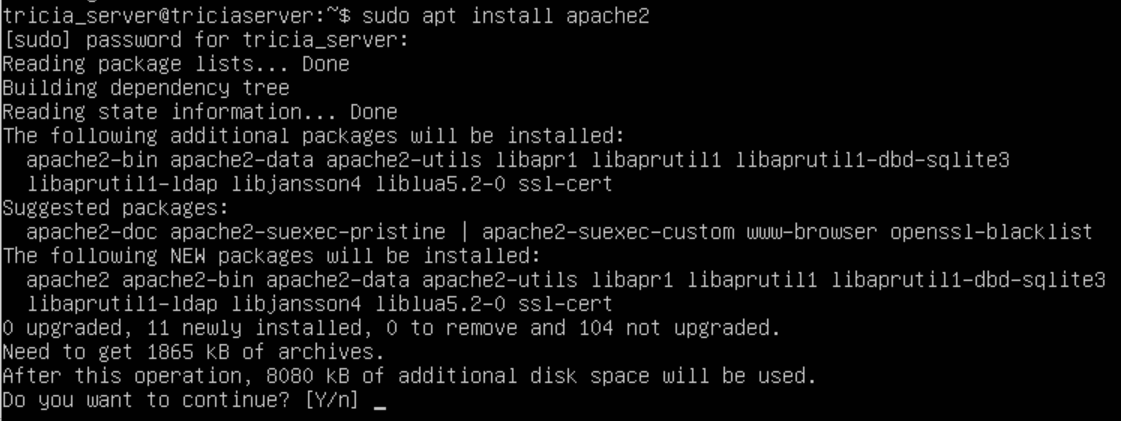
Ssh tricia\_server



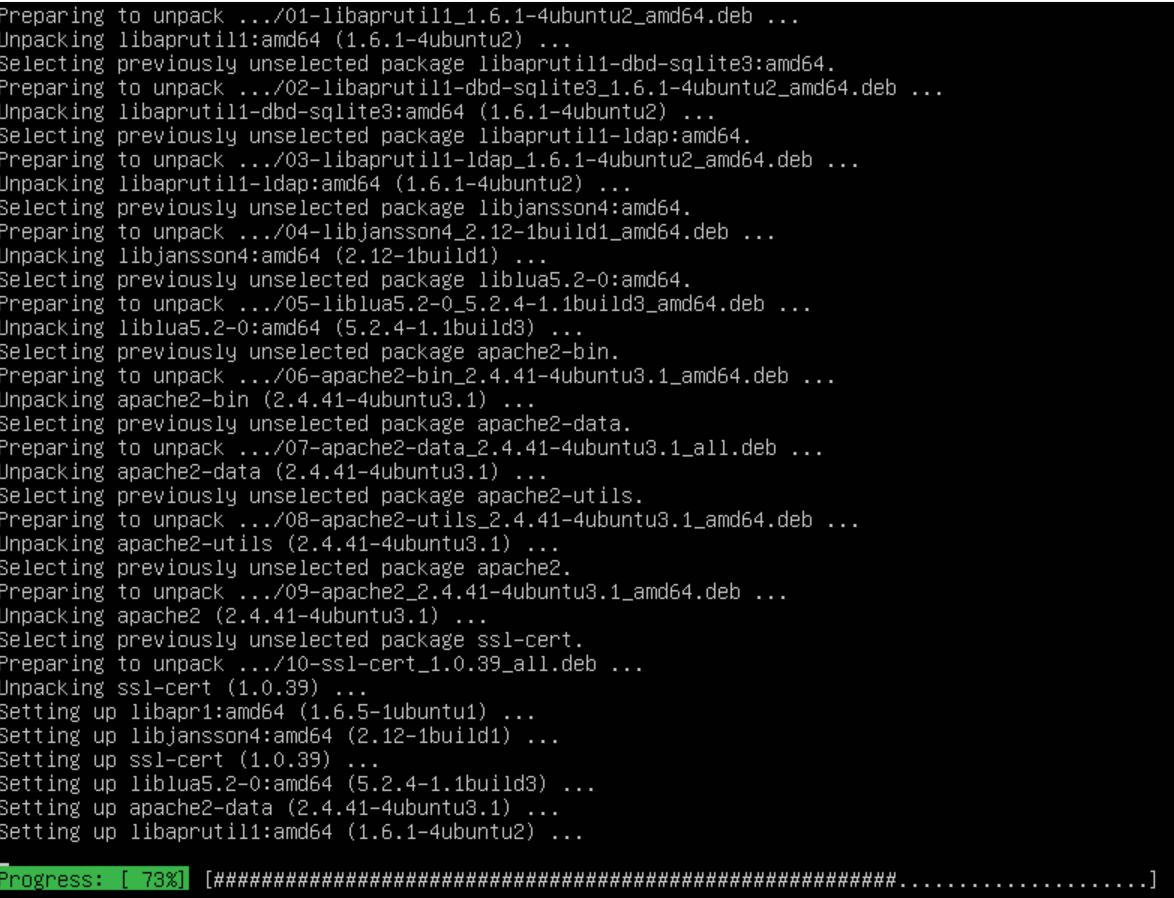
## Instalando Apache

Usamos el siguiente comando para instalar Apache en nuestro servidor Ubuntu

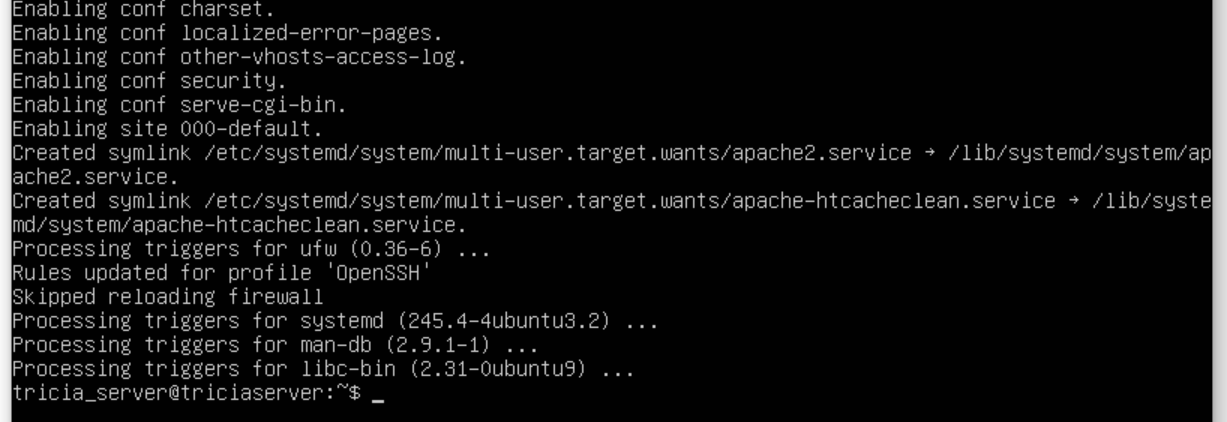
Sudo apt install apache2



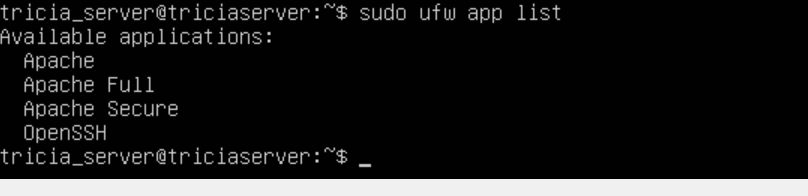
Empieza la instalación tras pulsar Y e intro:



Y acaba:



Comprobamos la instalación con **sudo ufw app list**



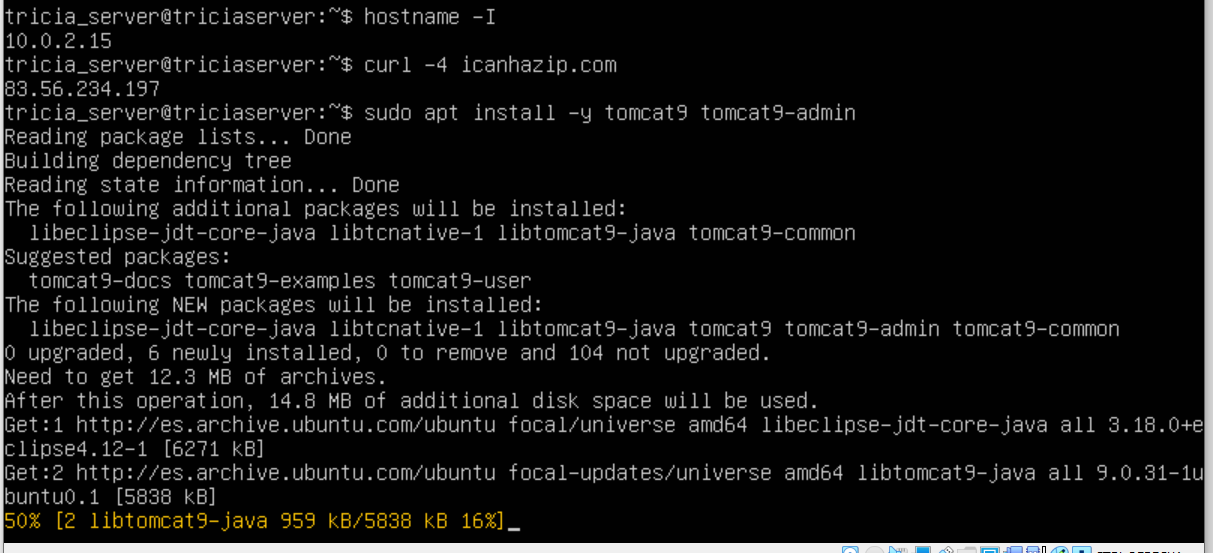
Vemos que nos aparecen 3:

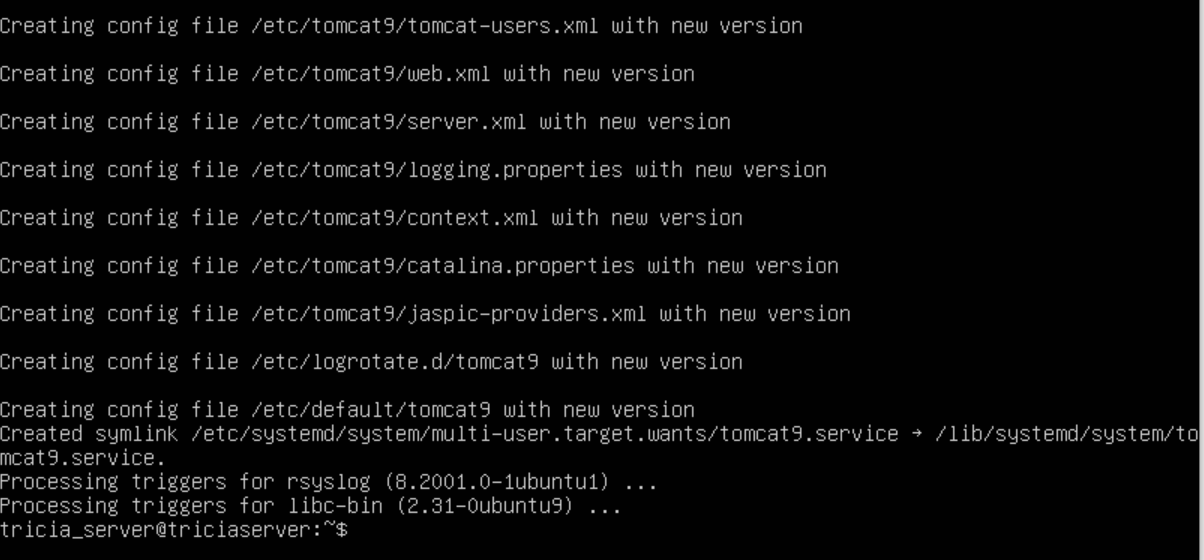
* **Apache**: este perfil abre solo el puerto 80 (tráfico web normal no cifrado)
* **Apache Full**: este perfil abre el puerto 80 (tráfico web normal no cifrado) y el puerto 443 (tráfico TLS/SSL cifrado)
* **Apache Secure**: este perfil abre solo el puerto 443 (tráfico TLS/SSL cifrado)

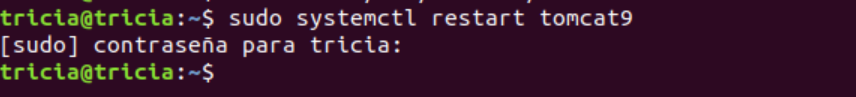
# Instalar Tomcat en servidor Ubuntu

El comando para instalar Tomcat en Ubuntu es el siguiente:

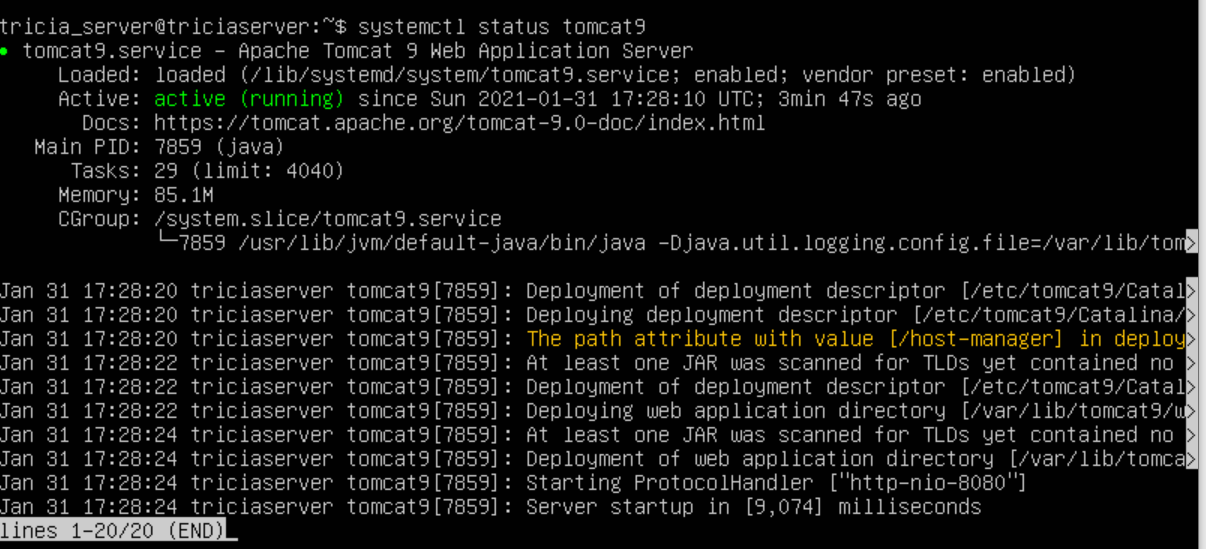
Sudo apt install -y tomcat9 tomcat9-admin







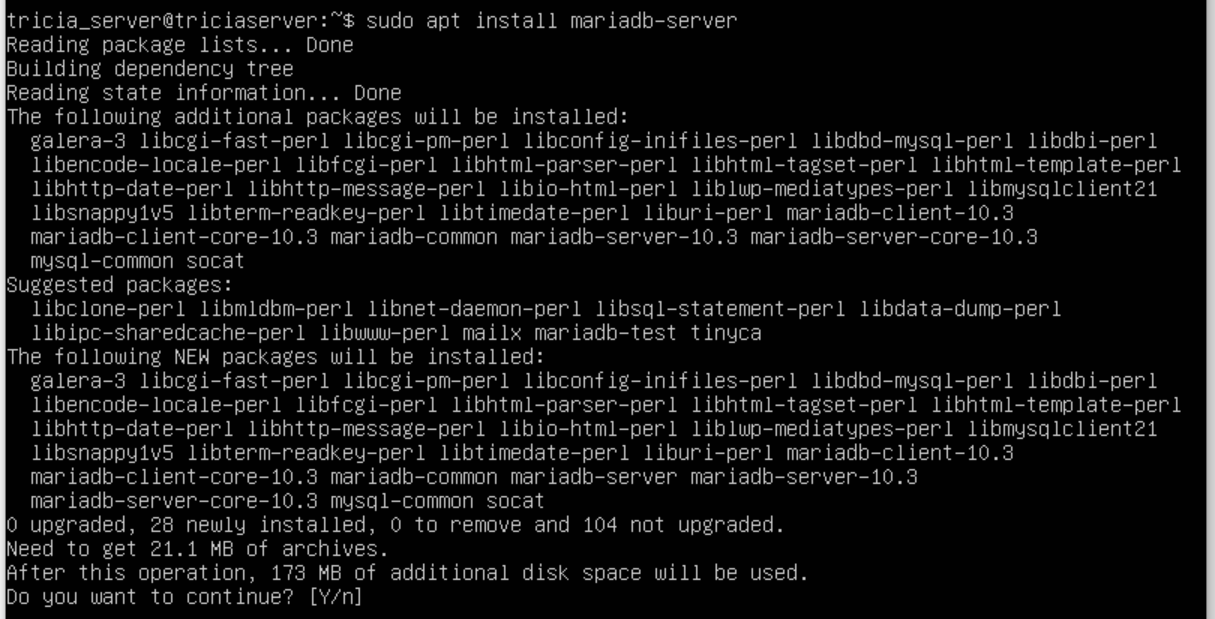
Una vez finaliza la instalación podemos comprobamos el estado de tomcat:



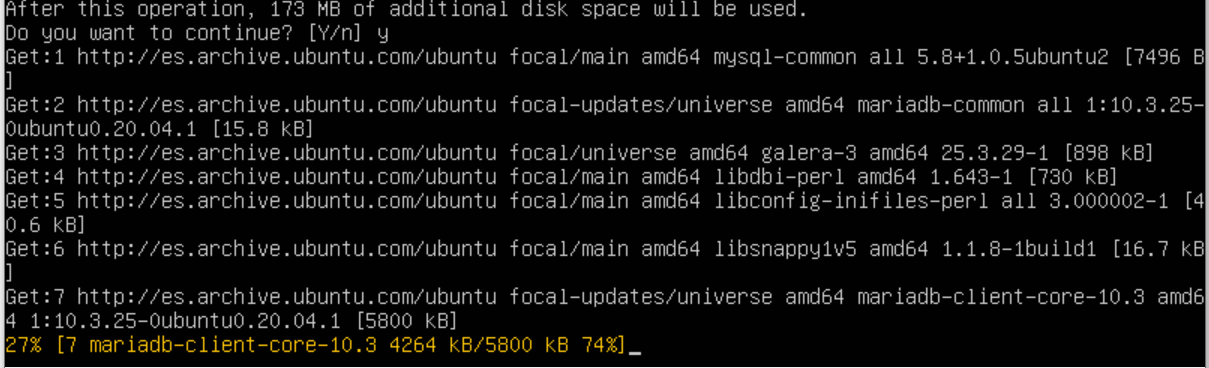
# Instalar MariaDB en servidor Ubuntu

La instalación se hace con:

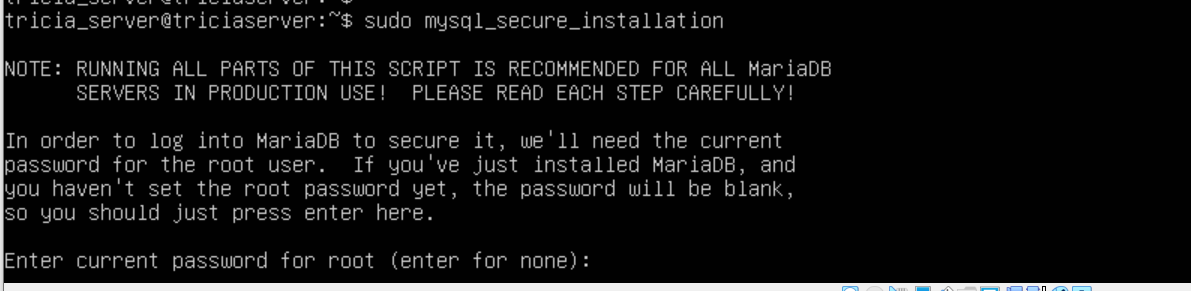
Sudo apt install mariadb-server



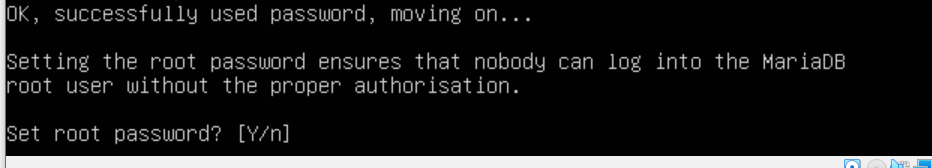
Pulsamos “Y” e intro y empezará la instalación:



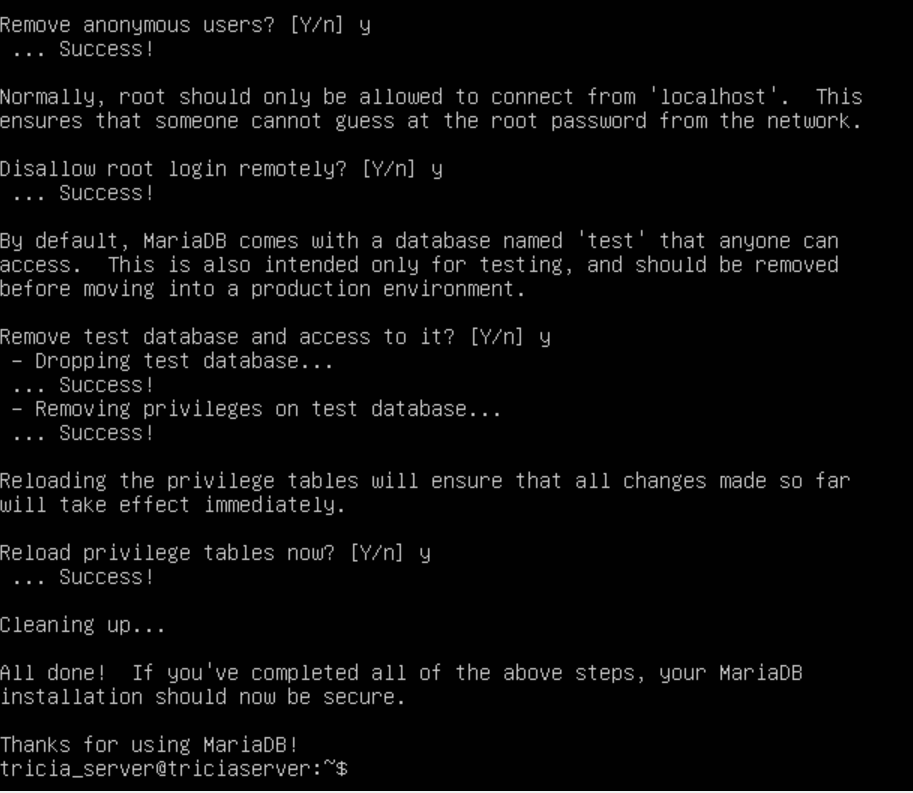
Una vez que ha terminado de instalarse podemos configurar su seguridad, es decir, restringir el acceso al servidor y eliminar cuentas no utilizadas. Para eso utilizamos un paquete que trae la propia mariadb: **sudo mysql\_secure\_installation**



Le damos a enter porque todavía no tenemos ninguna.



Decimos que N y enter. A partir de aquí le daremos Y y enter a todo para borrar usuarios anónimos, deshabilitar el poder hacer login con root en remoto, borrar la base de datos de test y su acceso y recarlar las tablas de permisos.



Ahora crearemos una cuenta para acceder usando contraseña sin tener que tocar el root, que en ocasiones puede ser peliagudo.

Sudo mariadb

Generamos el usuario con los siguientes comandos de mariadb:

GRANT ALL ON \*.\* TO ‘admin’@’localhost’ IDENTIFIED BY ‘1234’ WITH GRANT OPTION;

Las instrucciones acaban en ; admin es el usuario y 1234 la contraseña.

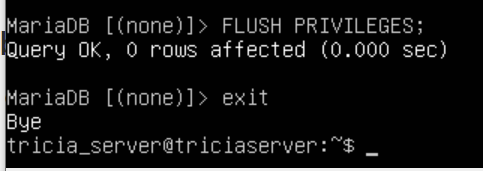


Lo siguiente será vaciar los privilegios para garantizar que se guarden y estén disponibles en la sesión actual y por último ya podremos cerrar el Shell de mariadb

FLUSH PRIVILEGES;

EXIT;

Copy

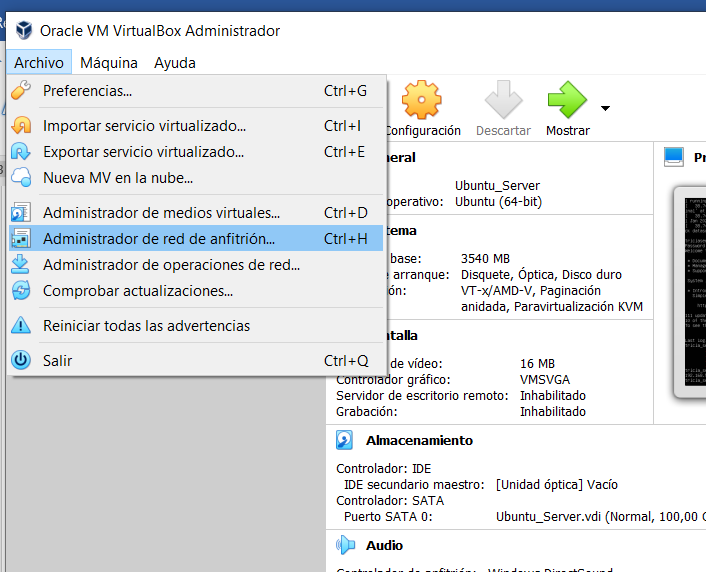


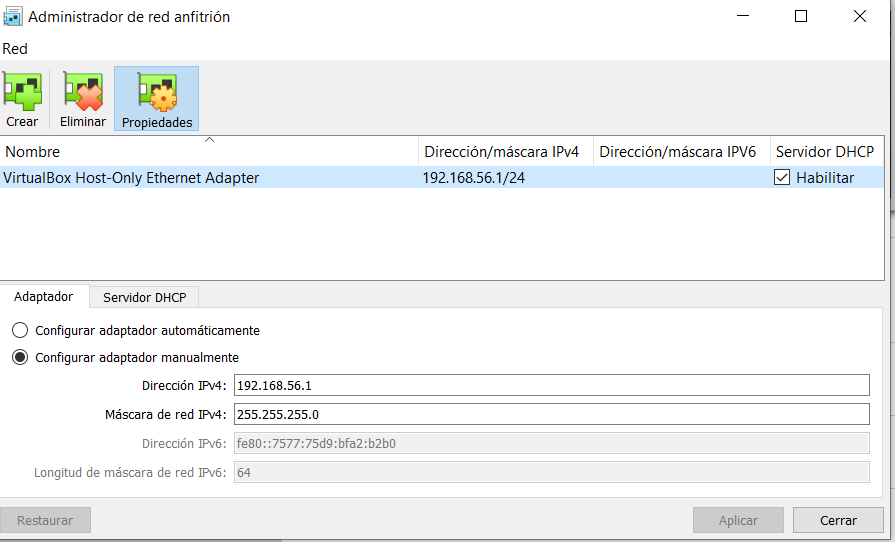
# Comprobación, configuración y accesibilidad

Una vez hemos instalado todos los paquetes requeridos, es hora de probar que la conexión con el servidor y los puertos para todas estas aplicaciones y servicios es la correcta.

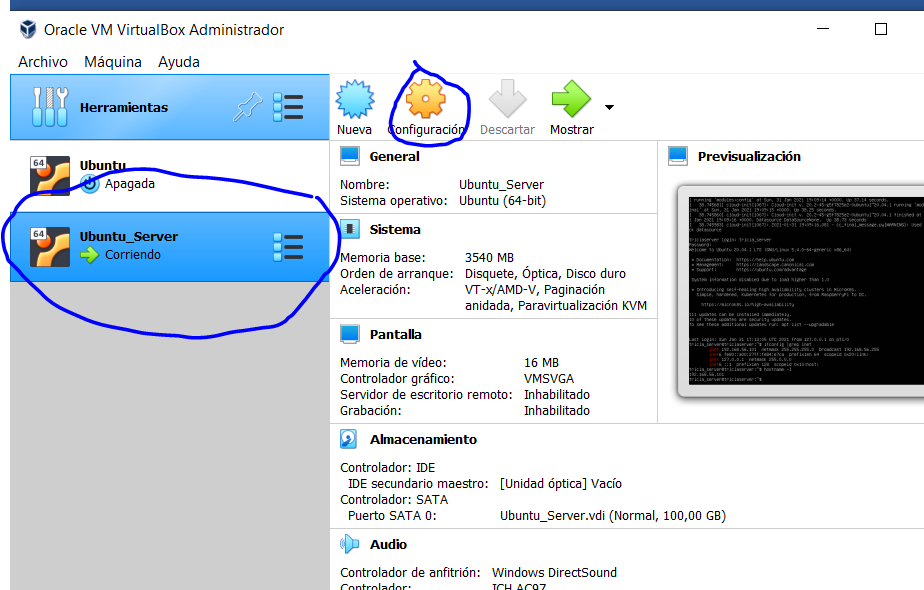
Si estamos usando un servidor Ubuntu instalado en una VirtualBox, hay que asegurarse de que la máquina virtual esté correctamente configurada:

Nos aseguramos que tenemos una red anfitrión:

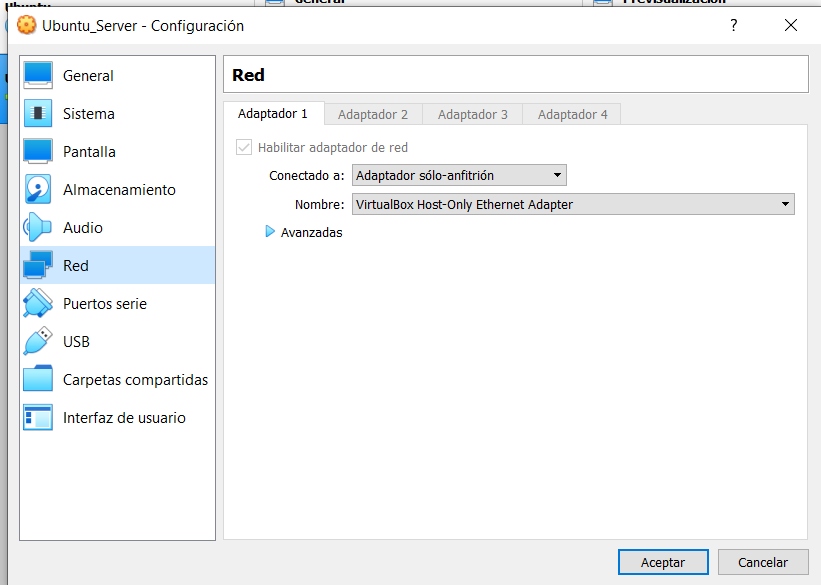




Y ahora debemos conectar dicha red a la máquina virtual de Ubuntu, para eso seleccionamos la que tiene nuestro servidor y damos a la opción “Configuración”:



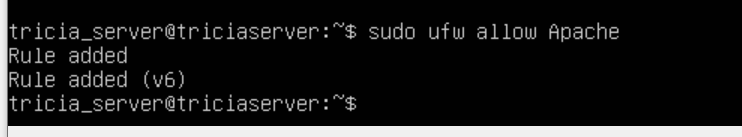
Navegamos a la opción Red y configuramos el Adaptador1 tal y como viene en la imagen:



## Apache

Antes de probar Apache, es necesario modificar los ajustes de firewall para permitir el acceso externo a los puertos web predeterminados. Por defecto se recomienda habilitar el más restrictivo que, en este caso, es Apache (puerto 80). Para ello usamos:

Sudo ufw allow Apache

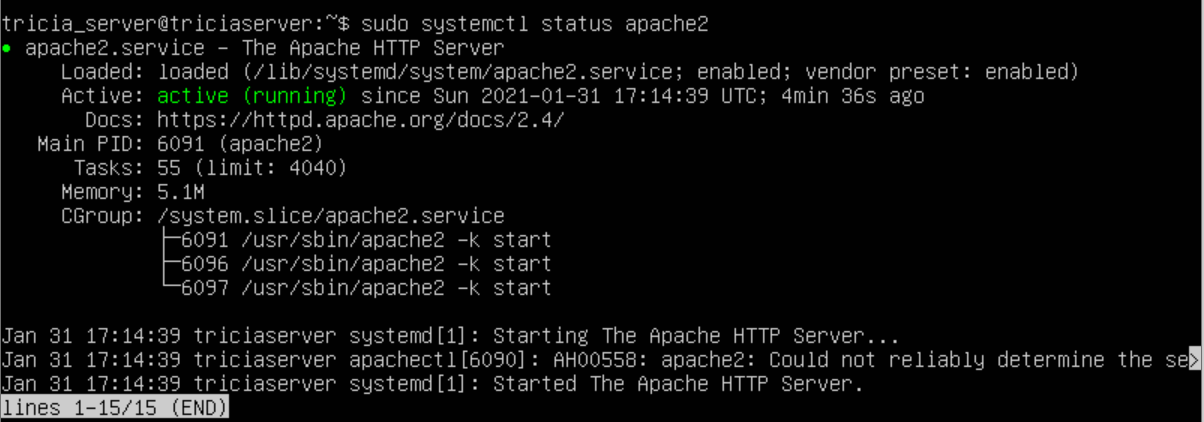


Y ya lo tendríamos: **sudo ufw status** lo comprobamos



Para ver si el servicio ya está en ejecución tecleamos:

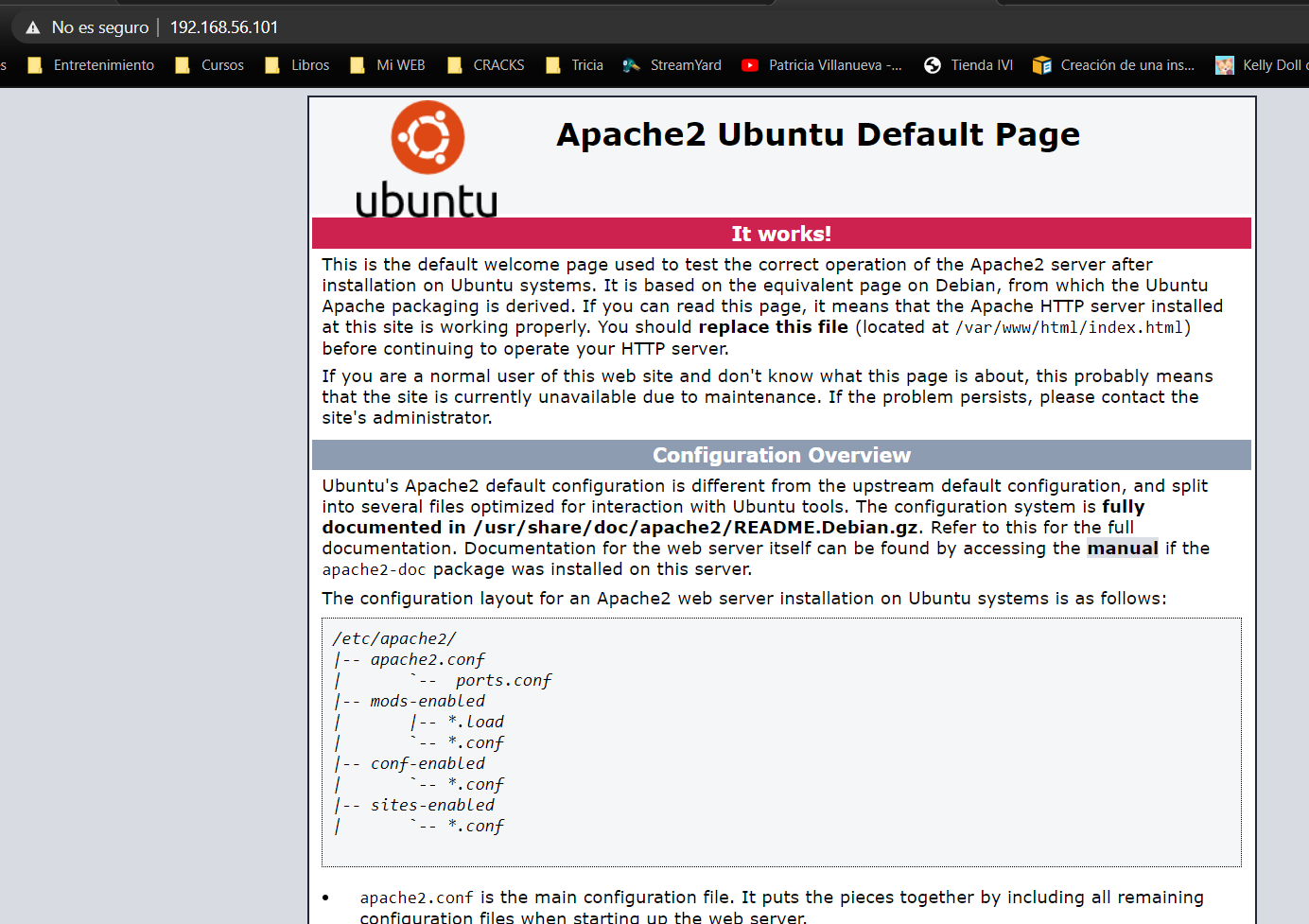
Sudo systemctl status apache2



Podemos ver que el servicio está activo y corriendo.

Otra forma es solicitar una página de Apache con el comando **hostname -I** y llamar a esa ip desde fuera.

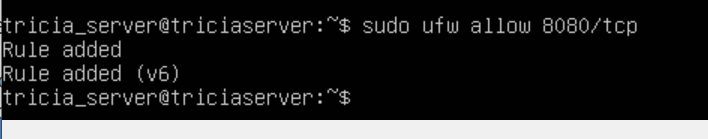




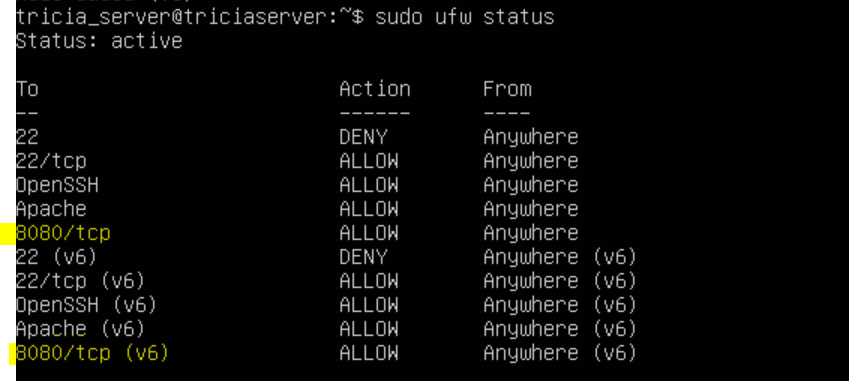
## Tomcat

Para poder acceder desde la red con el firewall instalado, debemos añadir lo siguiente:

Sudo ufw allow 8080/tcp



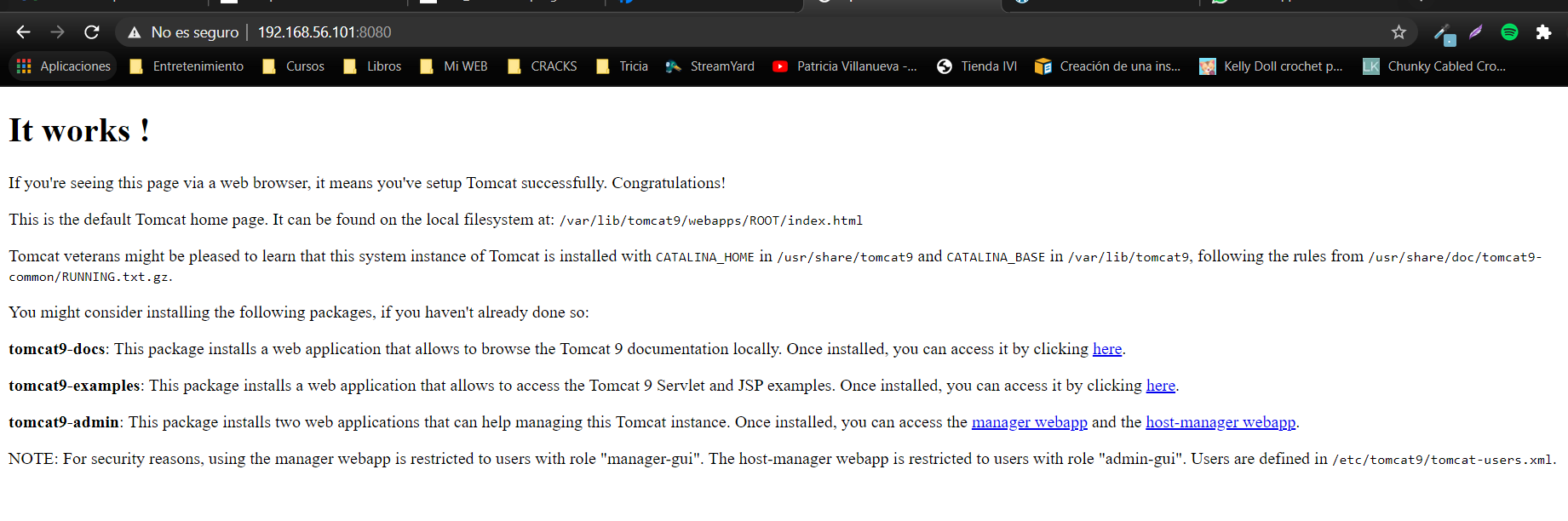
Comprobamos la conexión:



Y efectivamente, nos muestra que el puerto 8080/tcp para Tomcat se permite el acceso.

Ahora vamos a ver si podemos acceder desde nuestro ordenador con Windows al servidor de Ubuntu.

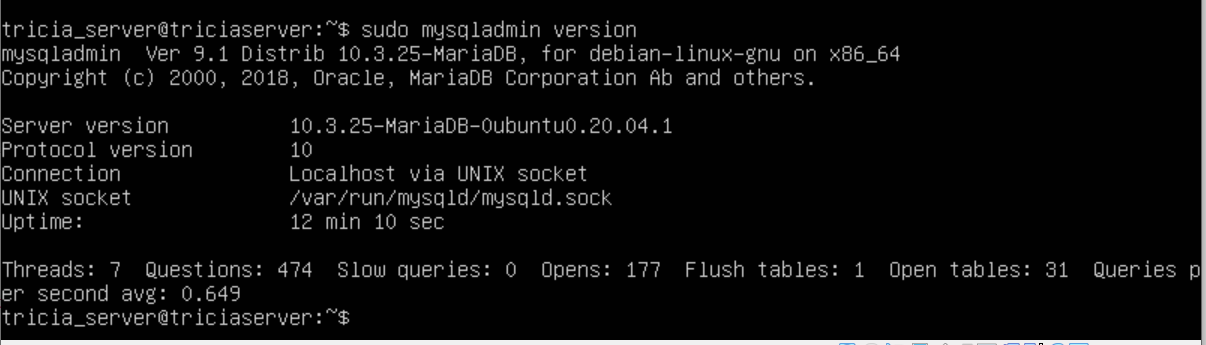
Para ello necesitamos la IP del servidor y el puerto de Tomcat



## Mariadb

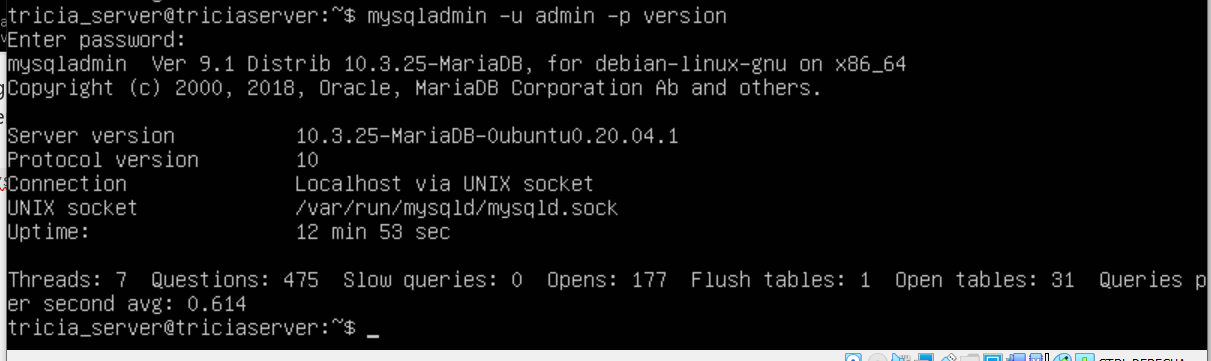
Aquí lo importante es que podamos acceder a la base de datos, para comprobar que esto es correcto, usaremos una herramienta de mariadb que es mysqladmin, que es un cliente que le permite ejecutar comandos administrativos.

sudo mysqladmin version



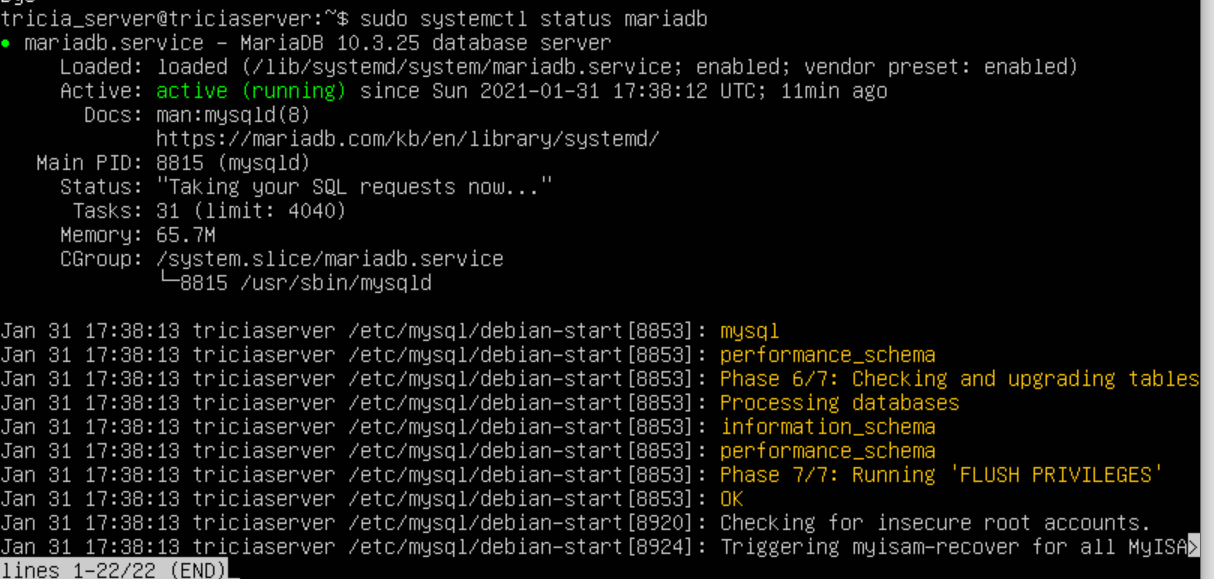
Si tengo un usuario administrativo independiente con la autenticación de contraseña, puedo realizar la misma operación escribiendo lo siguiente:

Mysqladmin -u admin -p version



Por último, probaremos la instalación de MariaDB.

Sudo systemctl status mariadb



Si MariaDB no funciona, puede iniciarla con el comando sudo systemctl start mariadb.