qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm

|  |
| --- |
| Configuración de Servidor.  Actividad 2. Tarea individual.  07/02/2021  Raúl Montero Peregrina |

ÍNDICE

[1.- Enunciado 3](#_Toc63691069)

[2.- Instalación y comprobación de servidores 4](#_Toc63691070)

[2.1.-JAVA 4](#_Toc63691071)

[2.2.-APACHE 5](#_Toc63691072)

[2.3.-TOMCAT 6](#_Toc63691073)

[2.4.-OPEN SSH 10](#_Toc63691074)

[2.5.-MARIADB 12](#_Toc63691075)

[3.- Comprobar y configurar puertos 14](#_Toc63691076)

[3.1.-APACHE 14](#_Toc63691077)

[3.2.-TOMCAT 16](#_Toc63691078)

[3.3.-MARIADB 18](#_Toc63691079)

[4.- Guía “How To” y subida a GitHUB 19](#_Toc63691080)

Actividad 2. Tarea Individual Configuración de Servidor



# 1.- Enunciado

**Requerimiento 1**

Queremos preparar nuestro servidor Linux para poder desplegar una aplicación web. Para ello tenemos que verificar que están instalados:

1. Java
2. Apache
3. Tomcat
4. openSSH
5. MariaDB

*Valoración: 6 puntos sobre 10*

**Requerimiento 2**

Así mismo, queremos asegurarnos de que los servidores están bien configurados y son accesibles antes de desplegar la aplicación. Por ello debemos configurar y comprobar que los puertos asociados a Apache, Tomcat y MariaDB están abiertos en el Firewall y son accesibles desde el exterior.

*Valoración: 4 puntos sobre 10*

**Consideraciones**

Para toda la actividad se valorará el orden y la claridad de la documentación, así como la facilidad de uso.

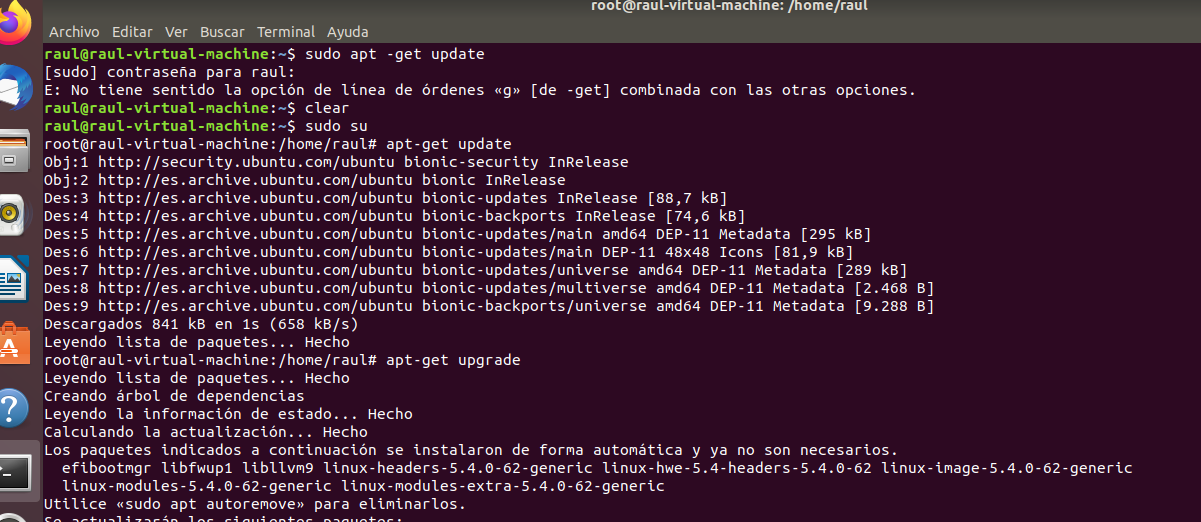
Para la entrega, es necesaria la creación y subida a la plataforma de un pequeño documento formal sobre la actividad (portada, explicación, etc.) y un Manual de instalación que describa y permita realizar todos los pasos para la instalación y configuración de las aplicaciones.

Nótese que más adelante se pedirá que se realicen tareas con un repositorio GIT que contenga la documentación de esta actividad.

# 2.- Instalación y comprobación de servidores

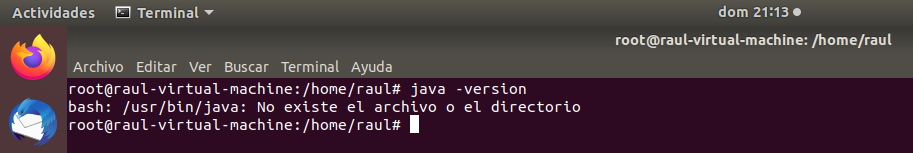
Para la realización de la actividad seguiremos utilizando nuestra máquina virtual de Ubuntu, utilizada para la actividad anterior.

Las instalaciones las realizaremos con el comando apt. Antes de proceder, comprobaremos si hay actualizaciones y en caso afirmativo procederemos a su instalación con el comando ***sudo apt –get update*** (busca y descarga actualización de paquetes) y ***sudo apt-get upgrade*** (instalas las actualizaciones descargadas anteriormente).Con el comando ***sudo su*** trabajaremos con las instalaciones como super usuario:

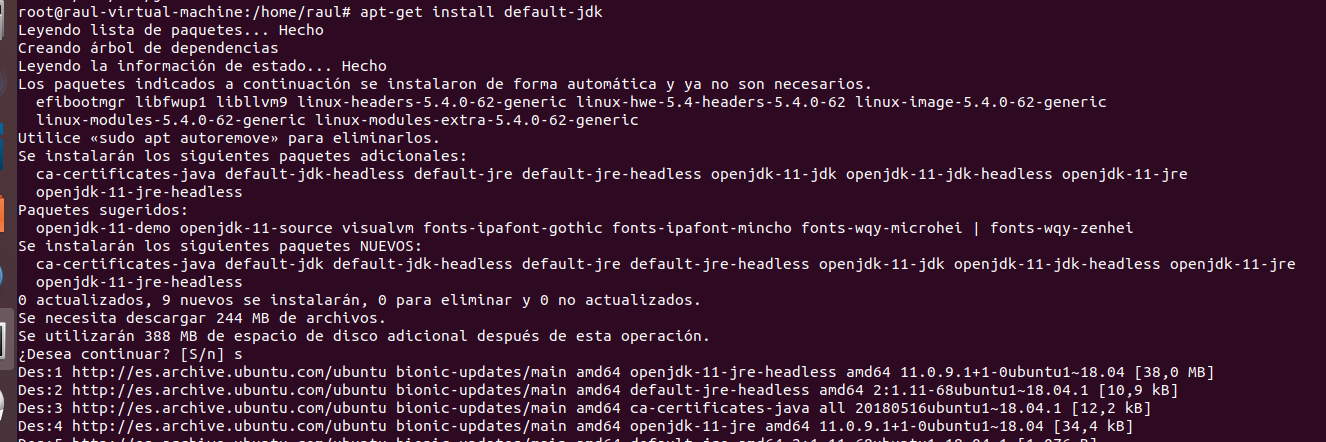


## 2.1.-JAVA

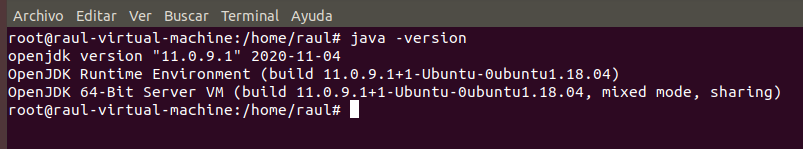
Para la instalación primero veremos con el comando ***java –version*** si ya tenemos java instalado:



Comprobado que no está instalado, con el comando ***apt-get install default-jdk***, procedemos a su instalación:

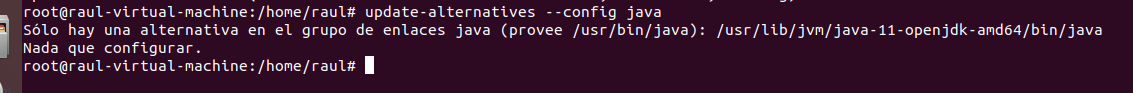


A continuación, con el comando ***java –version***, comprobaremos que se ha instalado correctamente volviendo a comprobar la versión de java instalada en nuestro SO Ubuntu:



Podemos ver la versión de java instalada y también la versión de Ubuntu que tenemos.

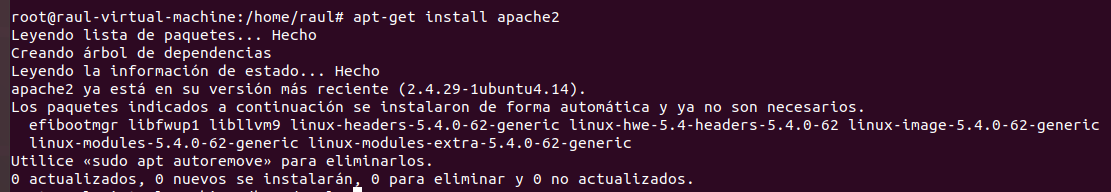
Para comprobar las instalaciones de Java que tenemos y configurar el JAVA\_HOME para indicarle la ruta, podemos ver las diferentes instalaciones y rutas de instalación con el comando ***update-alternatives --config java:***



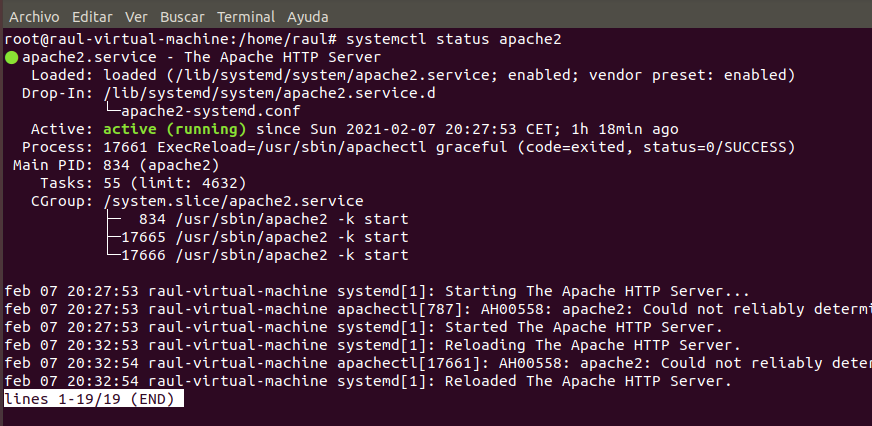
Como se puede observar solo tenemos una versión de java instalada y no sería necesario configurar el JAVA\_HOME para indicarle una ruta de instalación predeterminada.

## 2.2.-APACHE

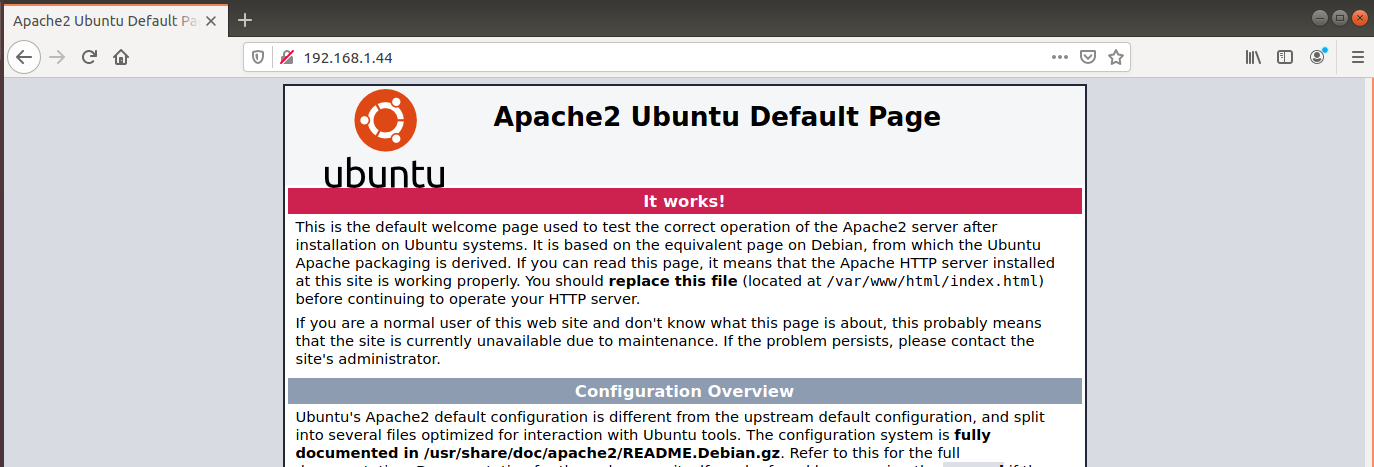
Para instalar Apache lo haríamos mediante el comando ***apt-get install apache2***. Al ejecutarlo veremos que nuestro sistema nos informa de que no se puede instalar al estar ya instalado y en la versión más reciente:



Podemos ver el estado de nuestro servidor corriendo, con el comando ***systemctl status apache 2***:

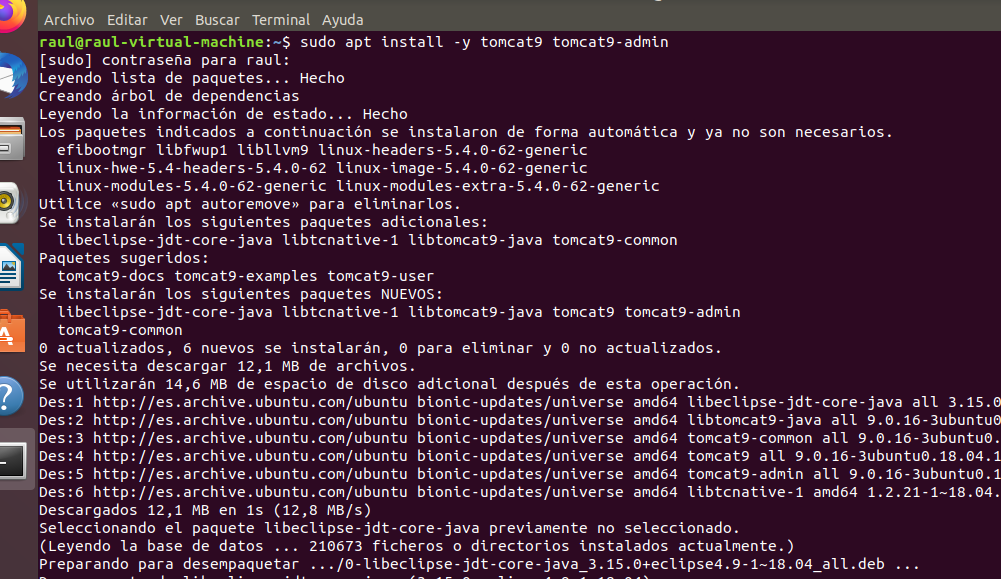


También podemos ver si está correctamente instalado tecleando en nuestro navegador nuestra IP. La obtenemos mediante el comando ***ifconfig*** y la insertamos en nuestro navegador:

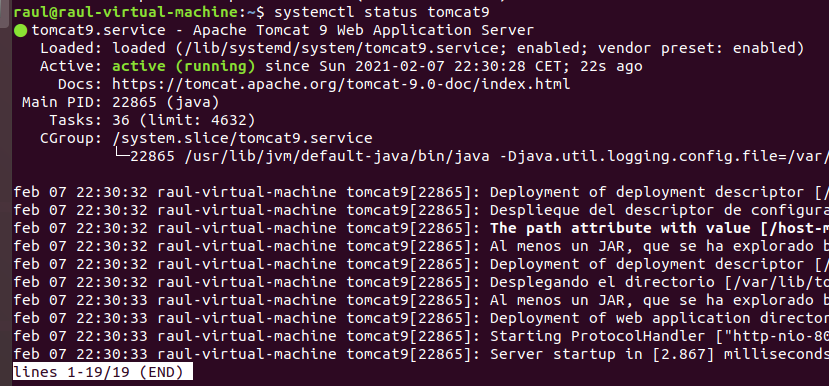


## 2.3.-TOMCAT

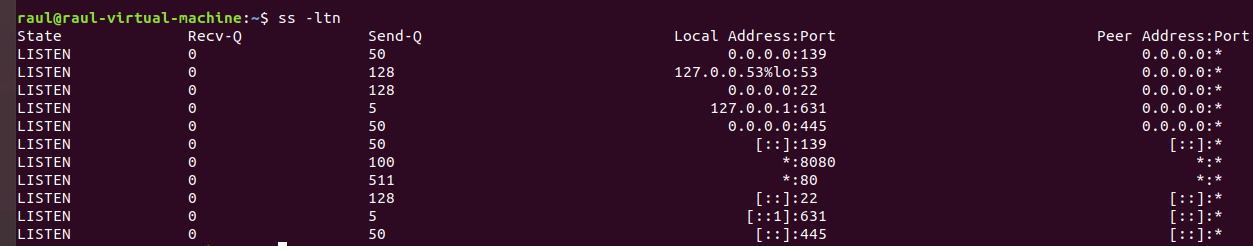
Instalamos Tomcat con el comando ***sudo apt install –y tomcat9 tomcat9-admin***:



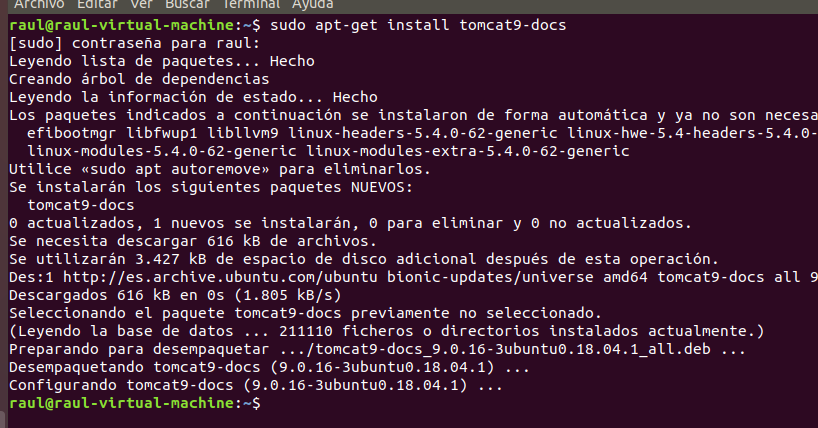
Después de la instalación comprobamos el estado con el comando ***systemctl status tomcat9***:

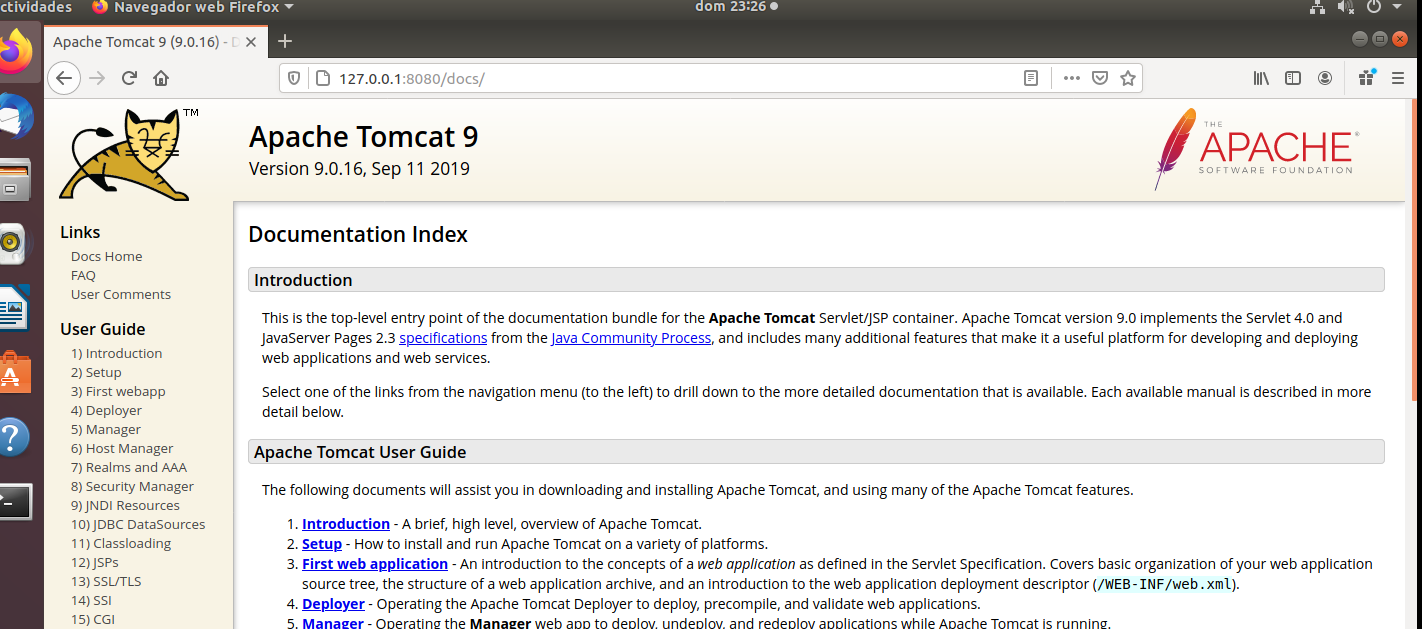


Con el comando ***ss -ltn*** podemos comprobar si nuestro puerto de Tomcat 8080 está escuchando, y comprobamos que sí y también el puerto de Apache 80:

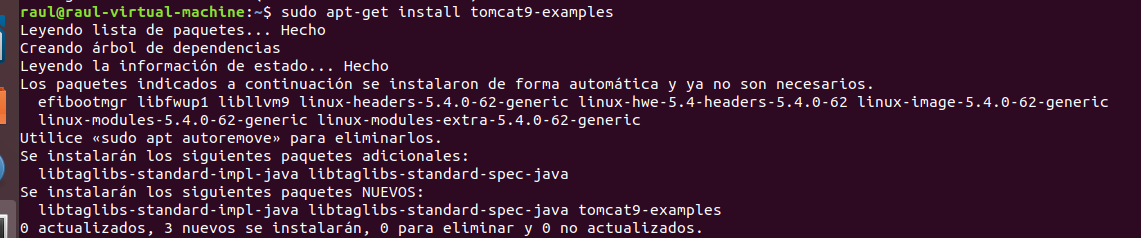


Posteriormente, instalaremos tomcat9-docs para ello usaremos el comando ***sudo apt-get install tomcat9-docs***:



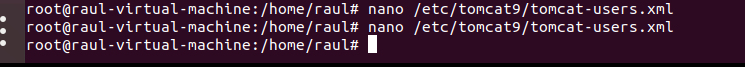


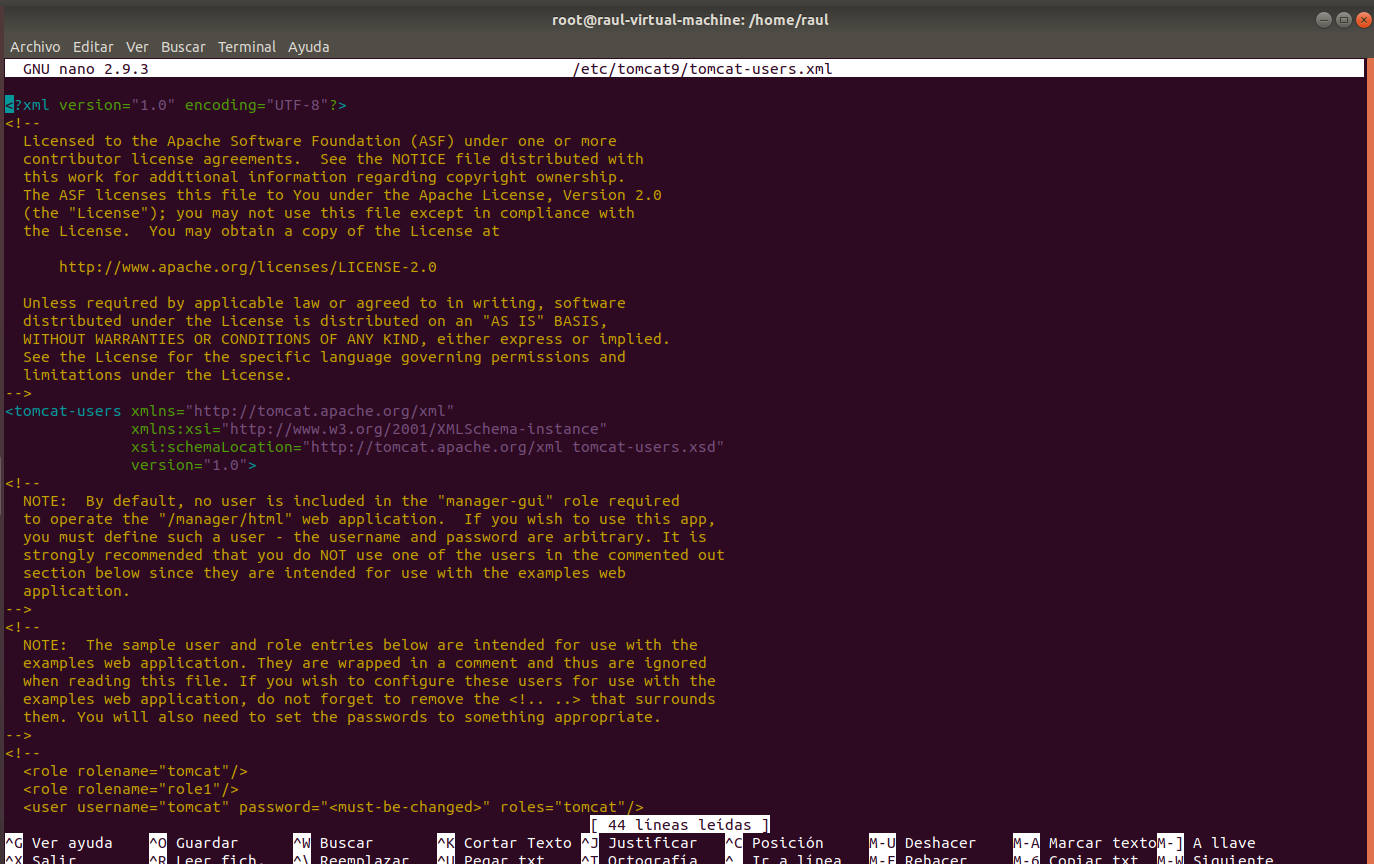
Instalaremos también los examples con el comando ***sudo apt-get install tomcat9-examples***:



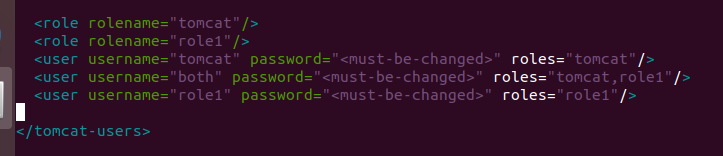


A continuación podremos acceder al manager web app, pero nos pedirá usuario y contraseña. Para ello tendremos que configurar nuestro usuario en el archivo xml correspondiente. Accedemos al archivo xml con comando ***nano /etc/tomcat9/tomcat-users.xml***:

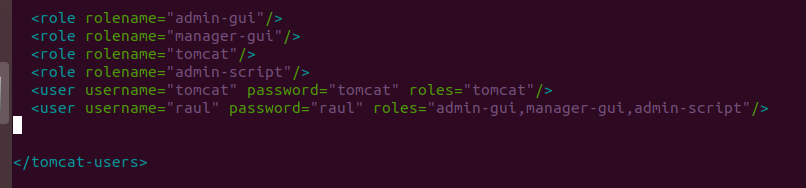




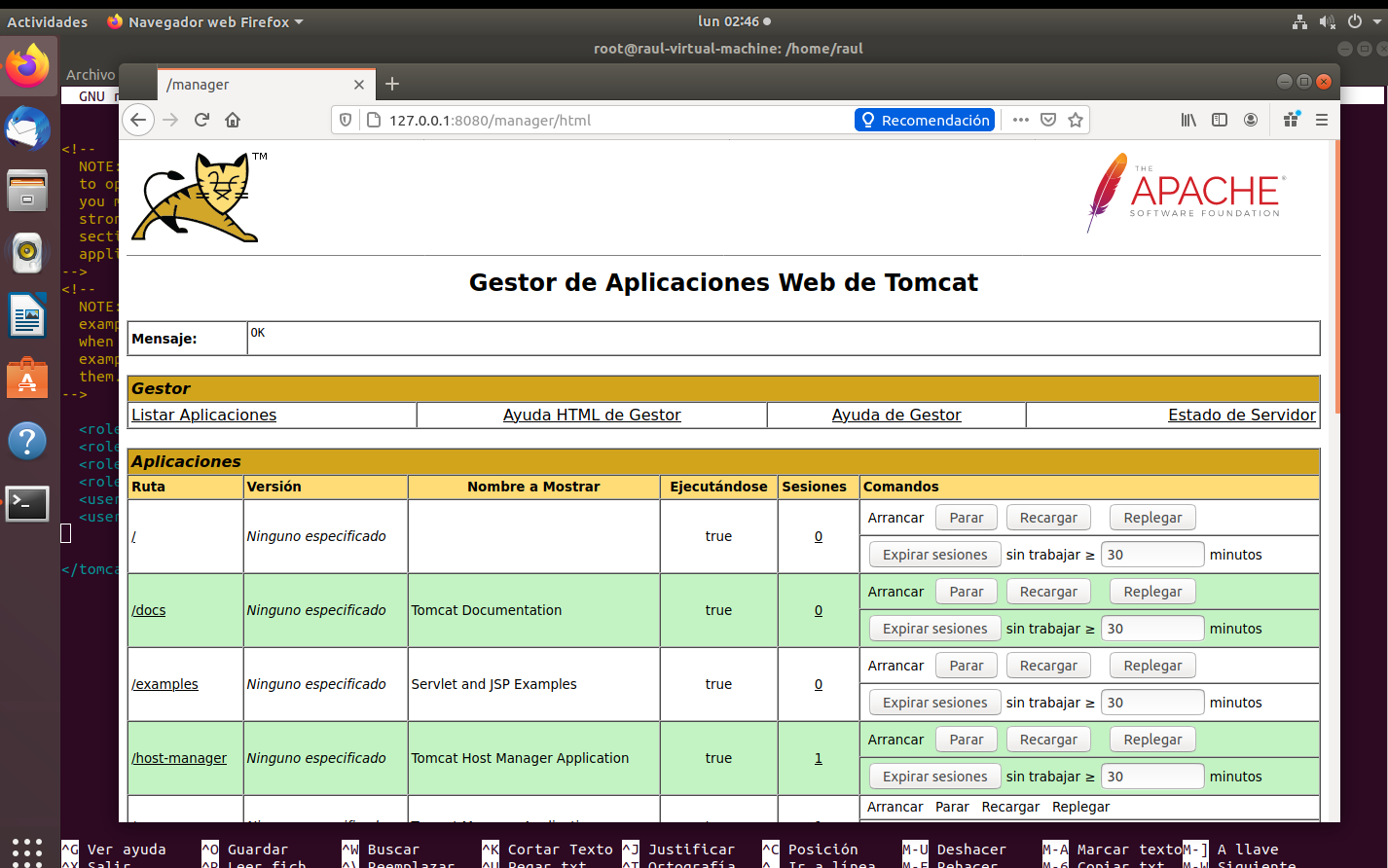
Aquí desmarcamos la parte de rolename y username....



....y añadimos los roles user-gui, admin-gui y admin-script con usuario y contraseña y le damos a guardar:

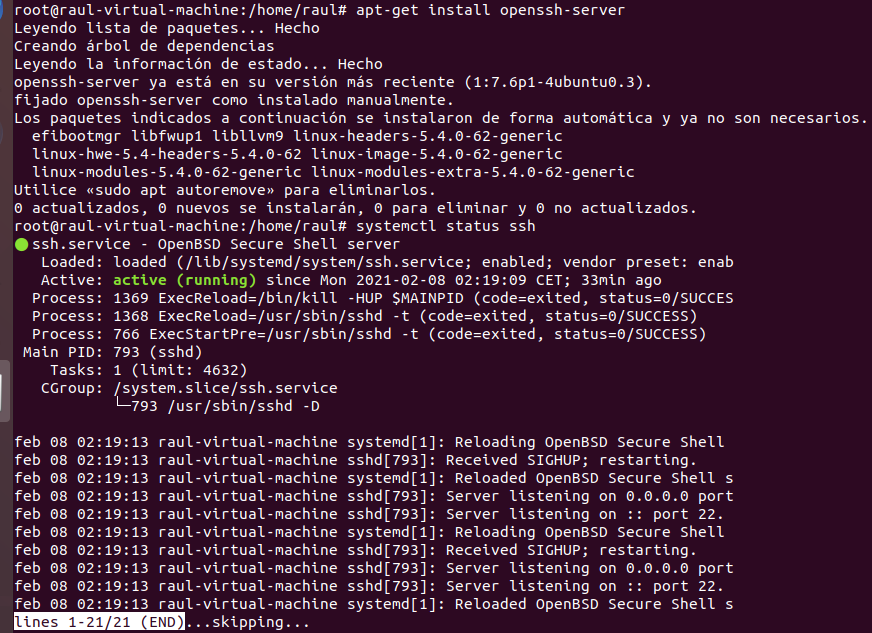


Reiniciamos el servicio Tomcat y accedemos de nuevo. Introduciendo nuestro usuario y contraseña vemos que ya nos deja acceder al manager aunque con una configuración muy básica:



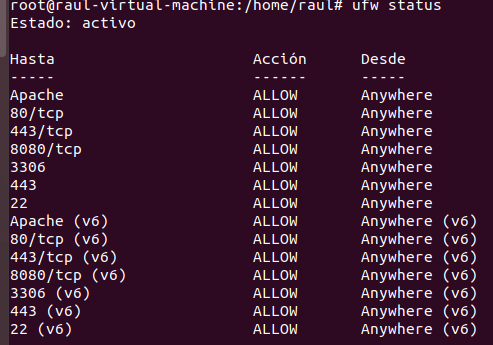
## 2.4.-OPEN SSH

SSH significa Secure Shell. Se trata de un protocolo de acceso remoto que nos proporciona un terminal seguro y encriptado, a través del puerto 22. **SSH** nos ofrece un terminal en una máquina cliente remota, desde la que podremos conectar a cualquiera de las cuentas que tenga el equipo remoto (que ha de tener activo el servicio de **servidor ssh**), con los privilegios que conlleve cada cuenta; así que podemos conectarnos con la cuenta root, con la cuenta de administrador, o con cualquier otra cuenta de usuario distinta .Para instalar SSH lo hacemos mediante el comando ***apt-get install openssh-server***:

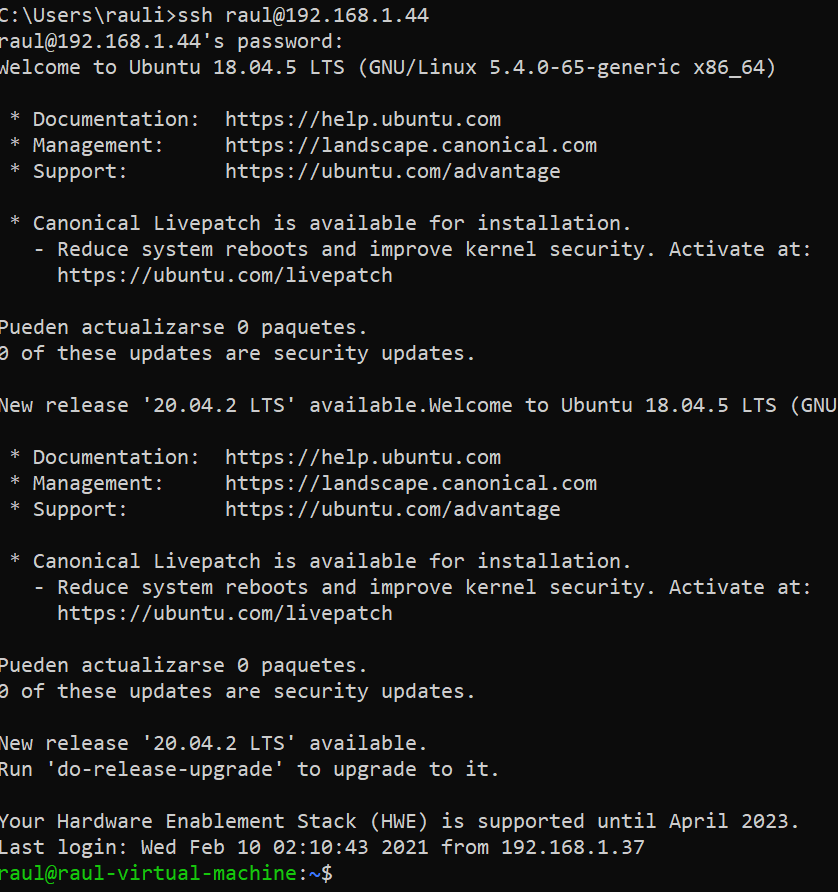


Como se puede observar ya estaba instalado y actualizado. Si comprobamos el estatus vemos que está activado y escuchando en el puerto 22, con el comando ***systemctl status ssh***.

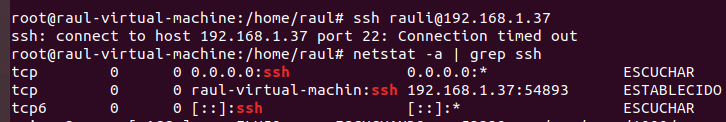
Comprobamos que el puerto 22 esté abierto el firewall con ***ufw status***:



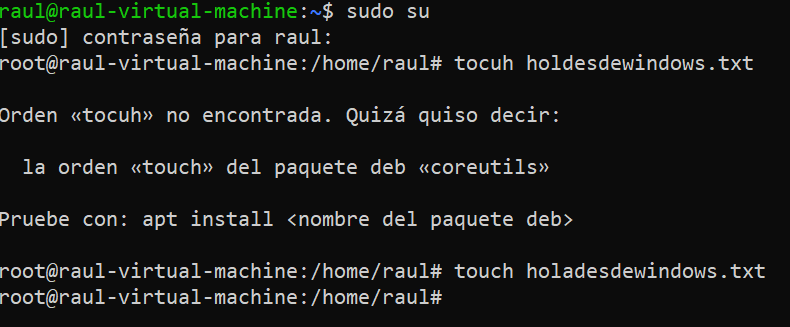
Ahora desde la Shell del sistema anfitrión con Windows, si tecleamos ***ssh usuario@ip*** deberíamos poder acceder a la Shell de Ubuntu en nuestra máquina virtual:

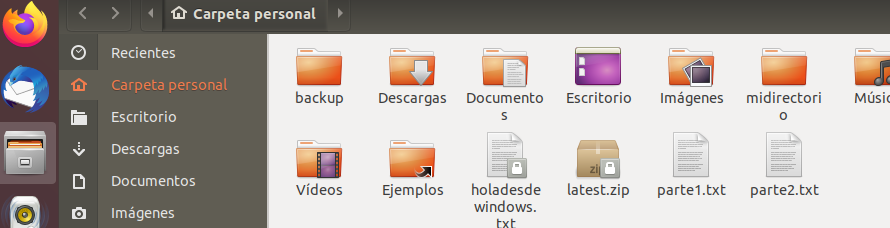


Si comprobamos el servicio ssh desde Ubuntu con el comando ***netstat –a | grep ssh***, observaremos que la IP del PC anfitrión está establecida:



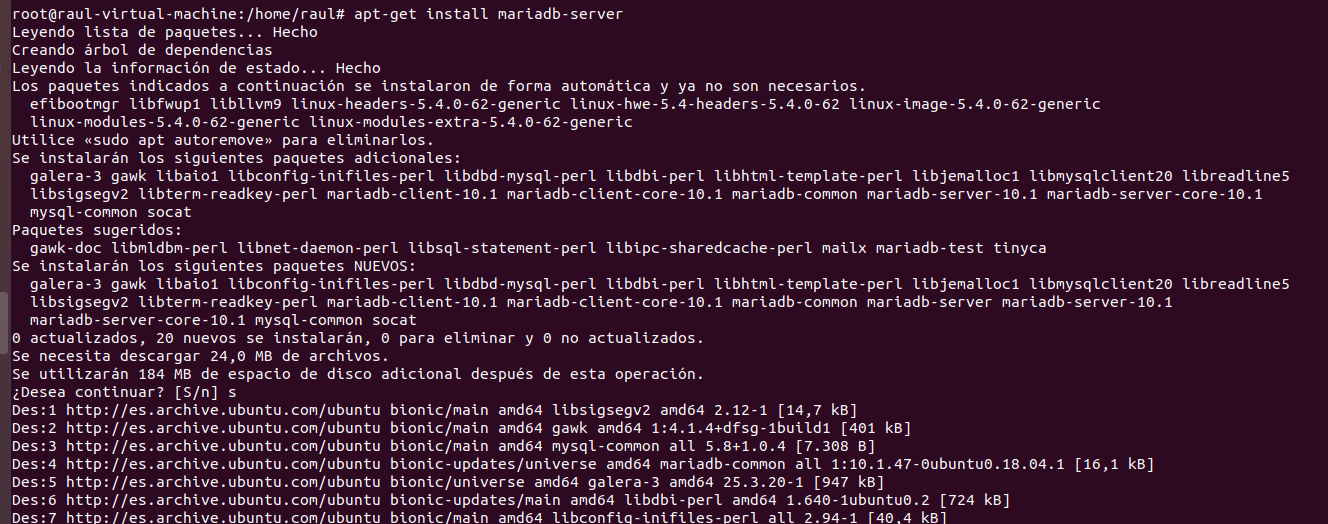
Si desde la Shell de Windows conectada a Ubuntu, creamos un archivo, veremos como en Ubuntu se ha creado, con lo que estamos conectados correctamente a la Shell de Ubuntu de la máquina virtual desde nuestro sistema anfitrión de Windows:



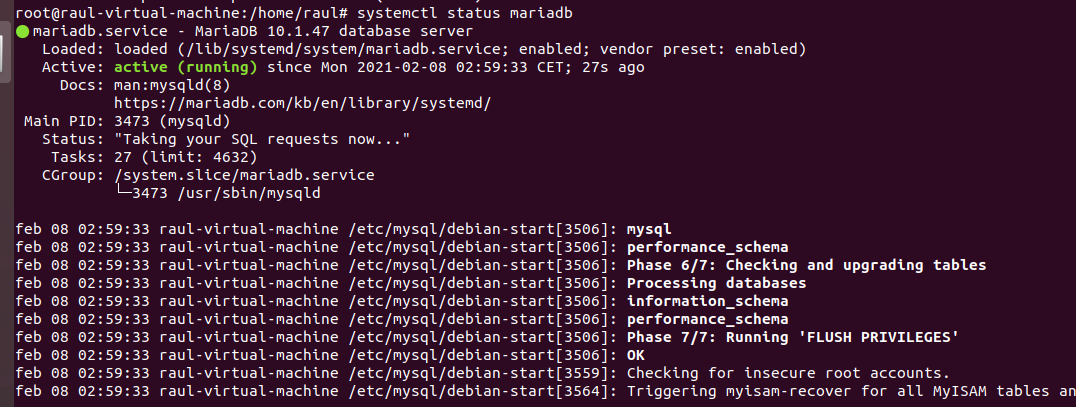


## 2.5.-MARIADB

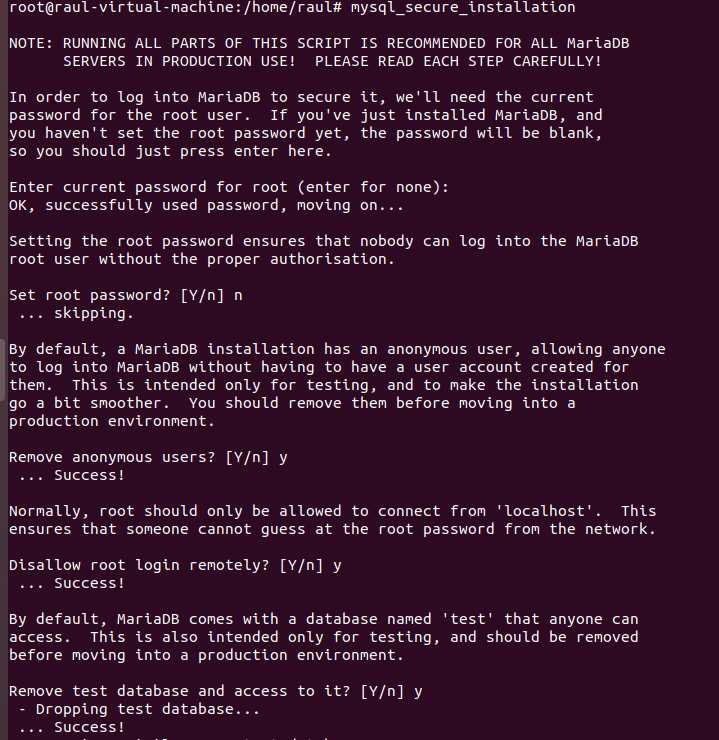
Mediante el comando ***sudo apt-get install mariadb-server*** podemos instalarlo:



Comprobamos si está activo y correctamente instalado con el comando ***sudo systemctl status mariadb***:

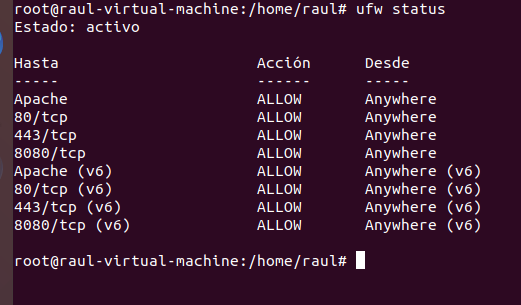


Realizaremos una pequeña configuración de Mariadb para quitar contraseñas y el resto lo dejamos igual, esto lo haremos con el comando ***mysql\_secure\_installation***:



# 3.- Comprobar y configurar puertos

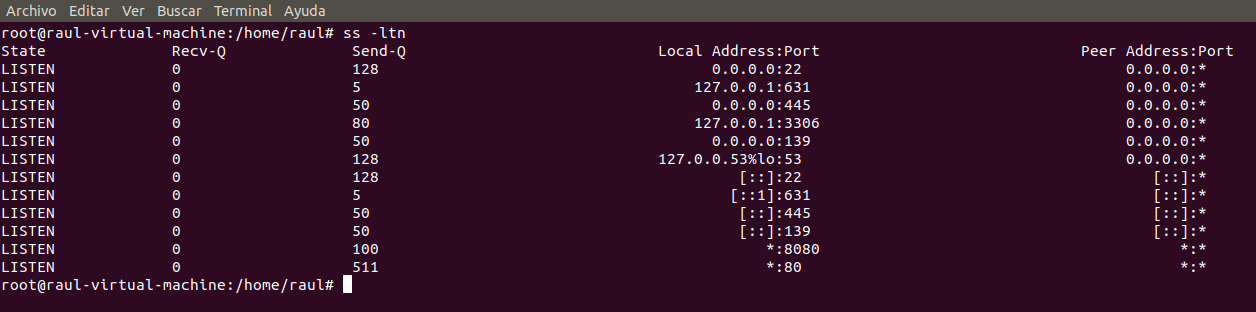
Para asegurarnos de que los servidores están bien configurados y son accesibles antes de desplegar la aplicación, configuraremos y comprobaremos que los puertos asociados a cada servidor están abiertos en el firewall y son accesibles desde el exterior. Antes de nada vamos a comprobar el estado de nuestro firewall con el comando ***ufw status***:



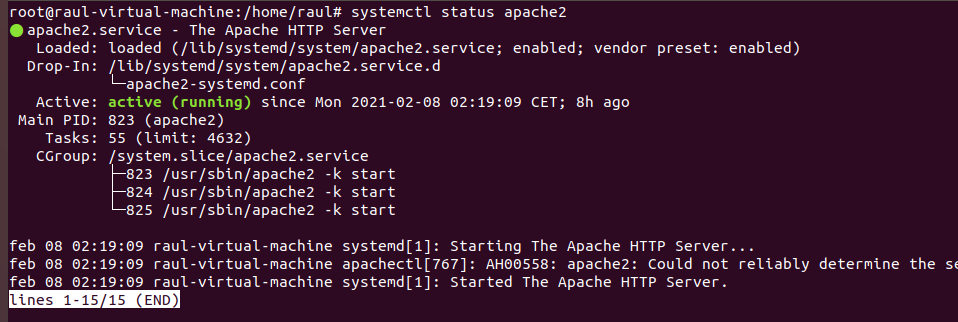
De ellos se desprende que todos los puertos asociados a nuestros servidores tienen permitido el tráfico a través de nuestro Firewal, menos el 3306 correspondiente a MaríaDB que no aparece. Crearemos la regla de permiso posteriormente:

## 3.1.-APACHE

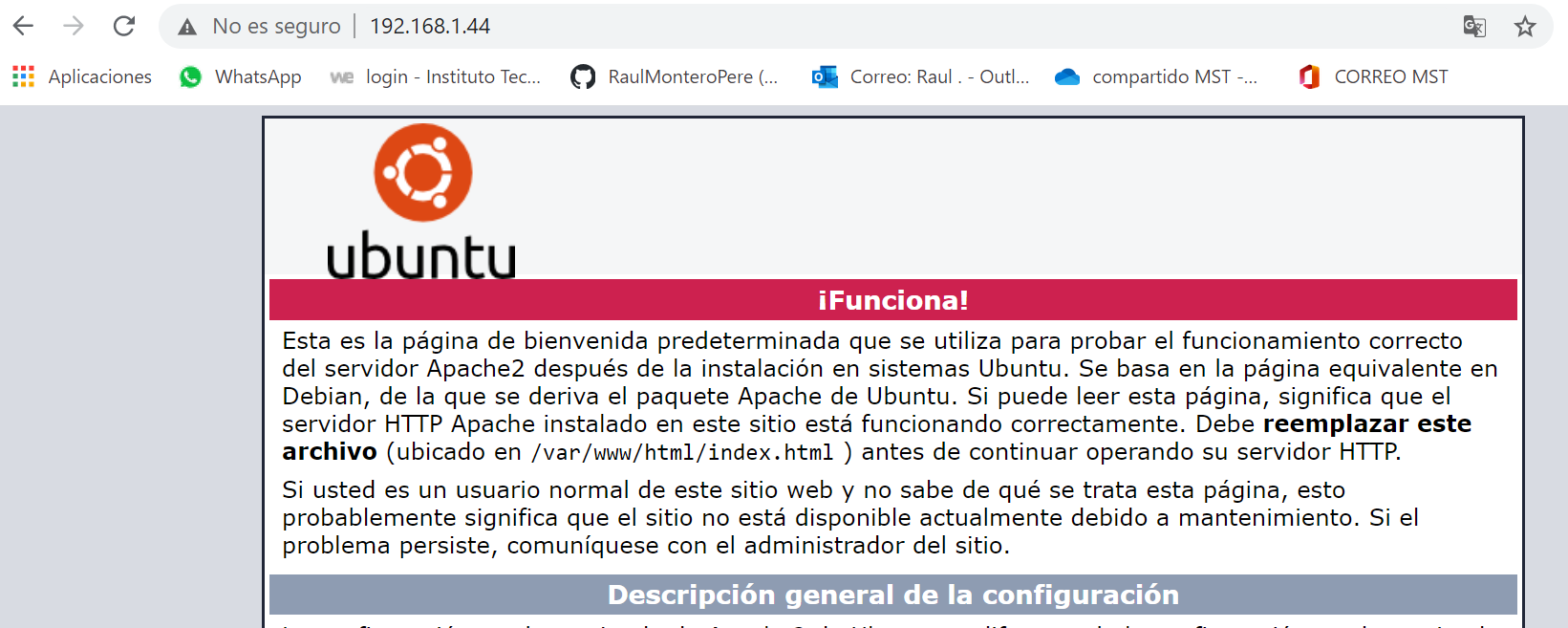
El puerto que utiliza Apache es el 80. Para comprobar que su puerto está escuchando correctamente ejecutaremos el comando ***ss –ltn***:



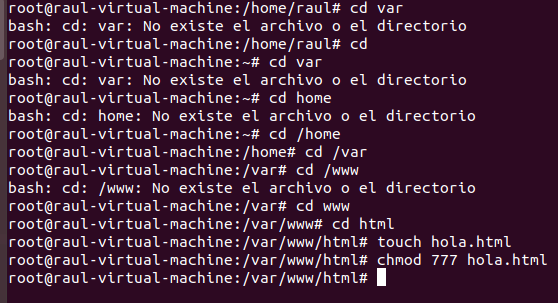
Con el comando ***systemctl status apache2*** veremos si nuestro servidor Apache está corriendo correctamente:

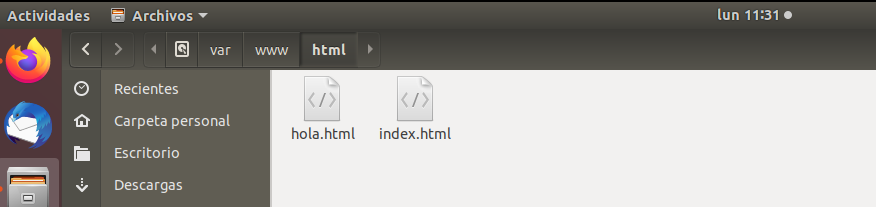


Por último verificaremos si acceso desde el exterior. Para ello desde nuestro PC anfitrión con SO Windows 10, accederemos al navegador y teclearemos la IP de nuestra máquina virtual con Ubuntu seguido del puerto 80 (192.168.1.44:80) y verificamos que nos responde la página de Apache:



Si por otro lado creamos un archivito HTML en el directorio de Apache var/www/html/hola.html con el comando touch desde la comandera, podemos hacerlo con ***cd /var/www/html*** primero nos dirigimos al archivo y una vez ahí con el comando ***touch hola.html*** creamos el archivo. Damos posteriormente todos los permisos a ese archivo para poder editarlo con ***chmod 777 hola.html***. Posteriormente si desde el navegador del sistema anfitrión tecleamos ***192.168.1.44:80/hola.html*** debería mostrarnos el contenido del html:





## 

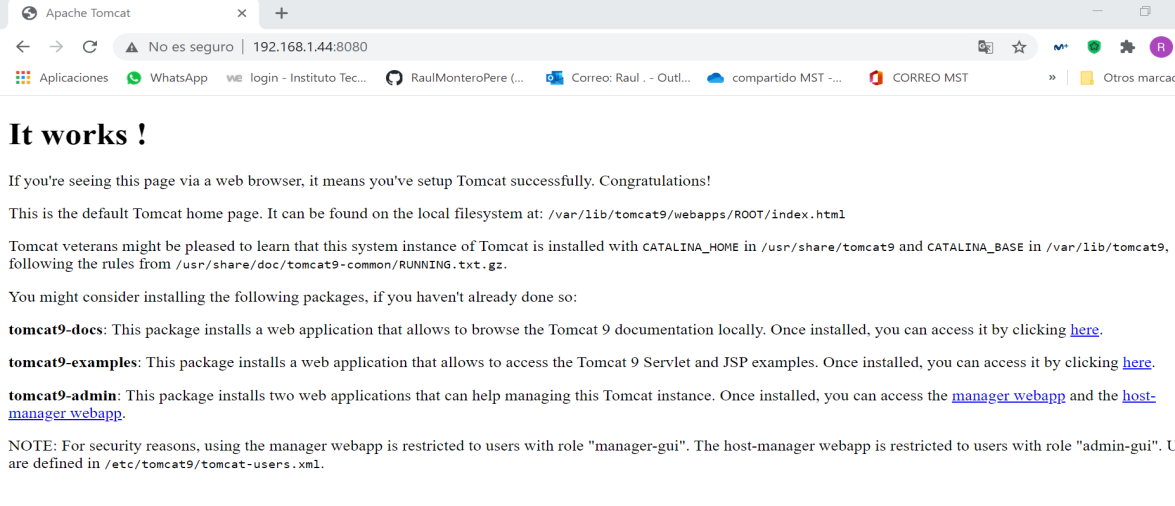
Verificamos que podemos acceder desde fuera a nuestro servidor APP Apache instalado en nuestra máquina virtual de Ubuntu.

## 3.2.-TOMCAT

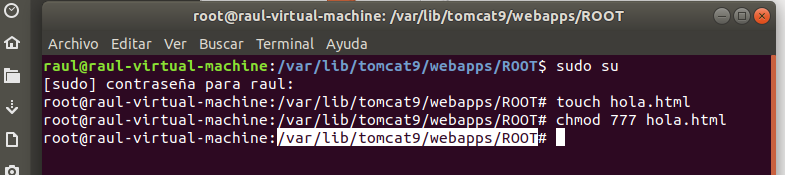
Como hemos comprobado anteriormente el puerto 8080 está escuchando y vamos a comprobar mediante el comando ***systemctl status tomcat9*** si el servicio está corriendo y activado:

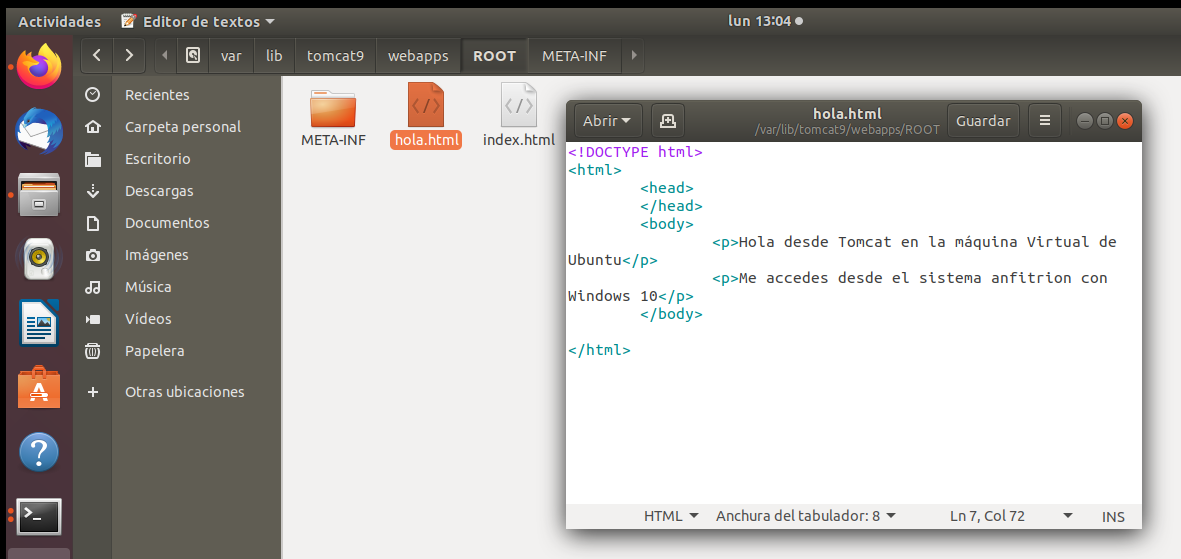


Para comprobar que tenemos acceso desde el exterior, si desde el sistema anfitrión tecleamos la IP de la máquina virtual y el puerto de Tomcat (192.168.1.8080) debería aparecernos la página de Tomcat, indicándonos así que hay acceso:

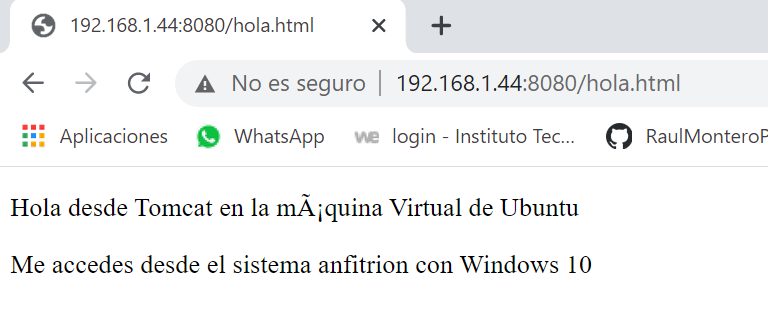


Por otro lado, los archivos .jsp, html y demás se guardan en Tomcat en el directorio var/lib/tomcat9/webapps/ROOT. Si abrimos una consola en esta ruta podemos crear un archivo .html escribiendo en la consola ***touch hola.html***. Posteriormente para dar permisos de escritura a este archivo, ejecutamos el comando ***chmod 777 hola.html***.



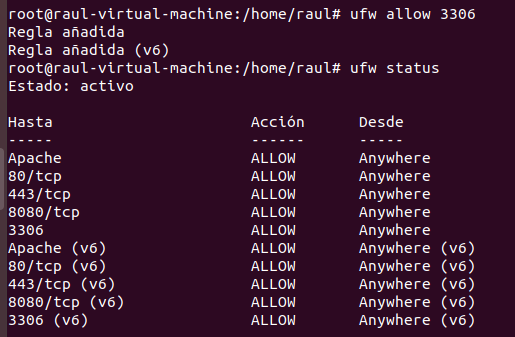


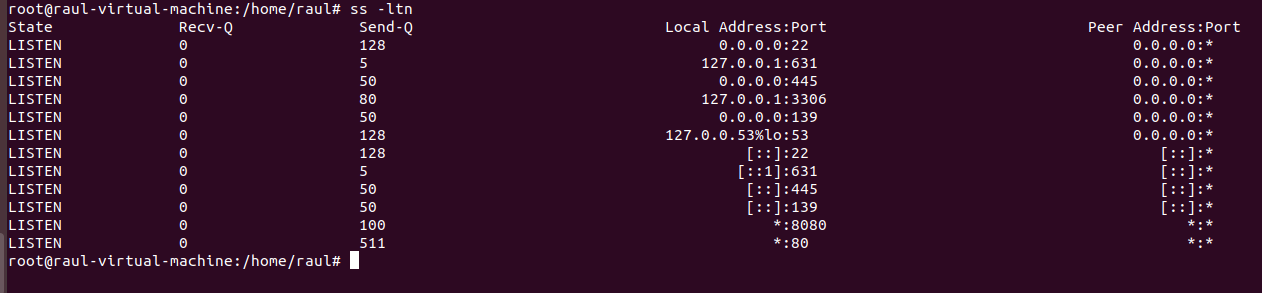
Si desde nuestro PC anfitrión con Windows 10 tecleamos en el navegador la IP, puerto y archivo html de Tomcat a visualizar, si Tomcat está funcionando en la máquina virtual, deberíamos de poder visualizarlo:



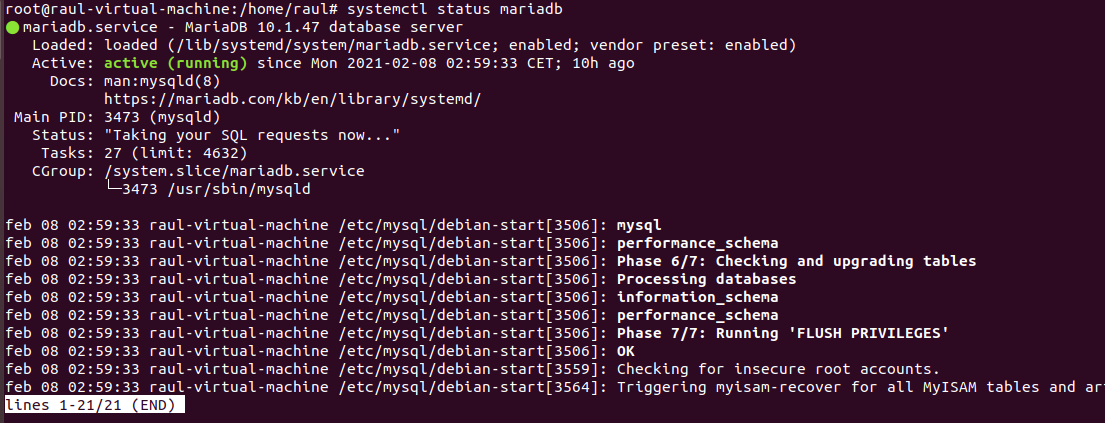
## 3.3.-MARIADB

Por último, vamos a abrir en nuestro firewall el puerto utilizado por MariDB que es el 3306, con el comando sudo ***ufw allow 3306***. Observamos con ***ufw status*** aquellos puertos y protocolos permitidos. A continuación comprobamos con el comando ***ss –ltn*** si el puerto está correctamente abierto y escuchando:





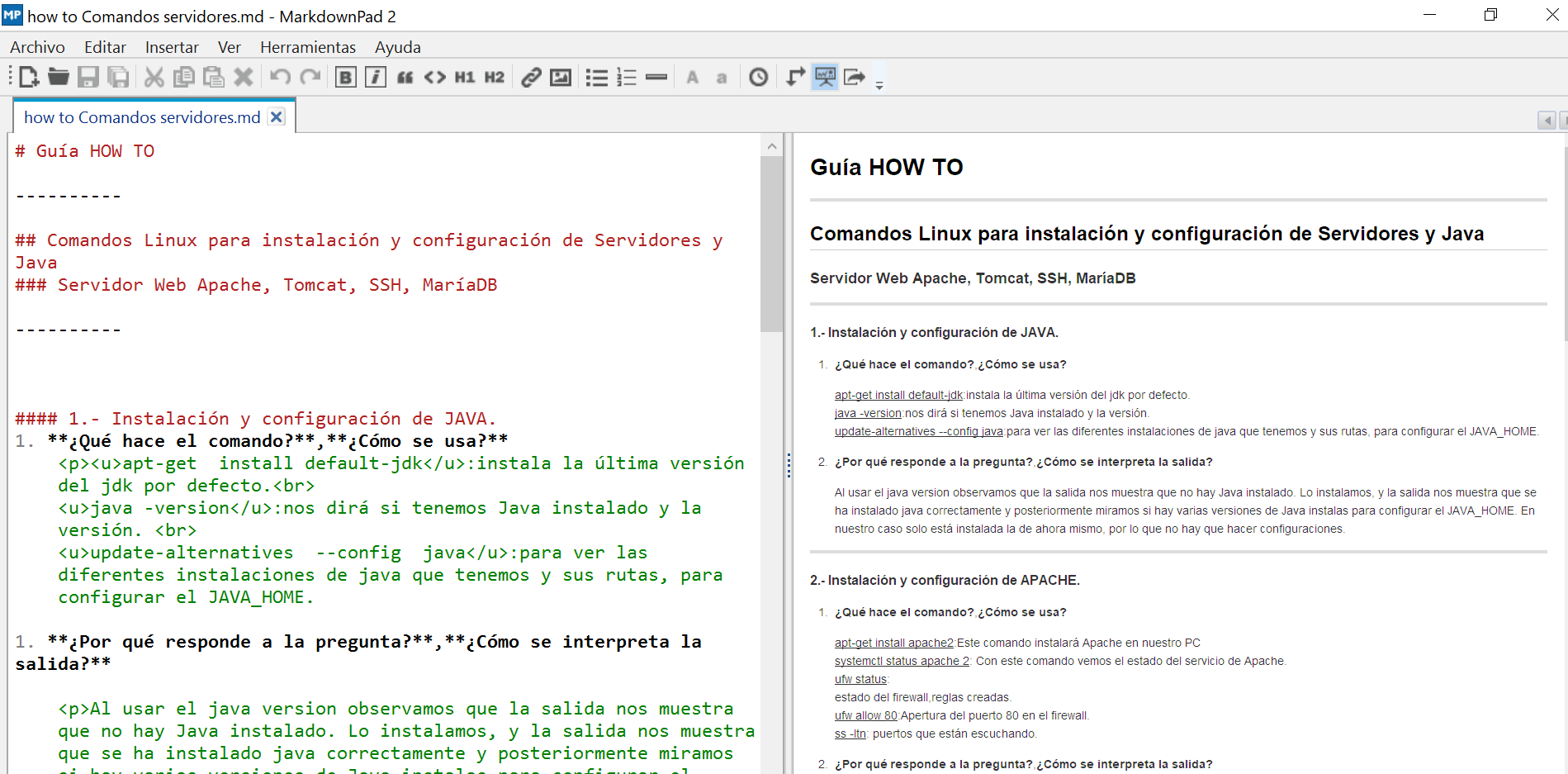
Con el comando ***systemctl status mariadb*** podemos comprobar también si el servicio está activado y corriendo correctamente:



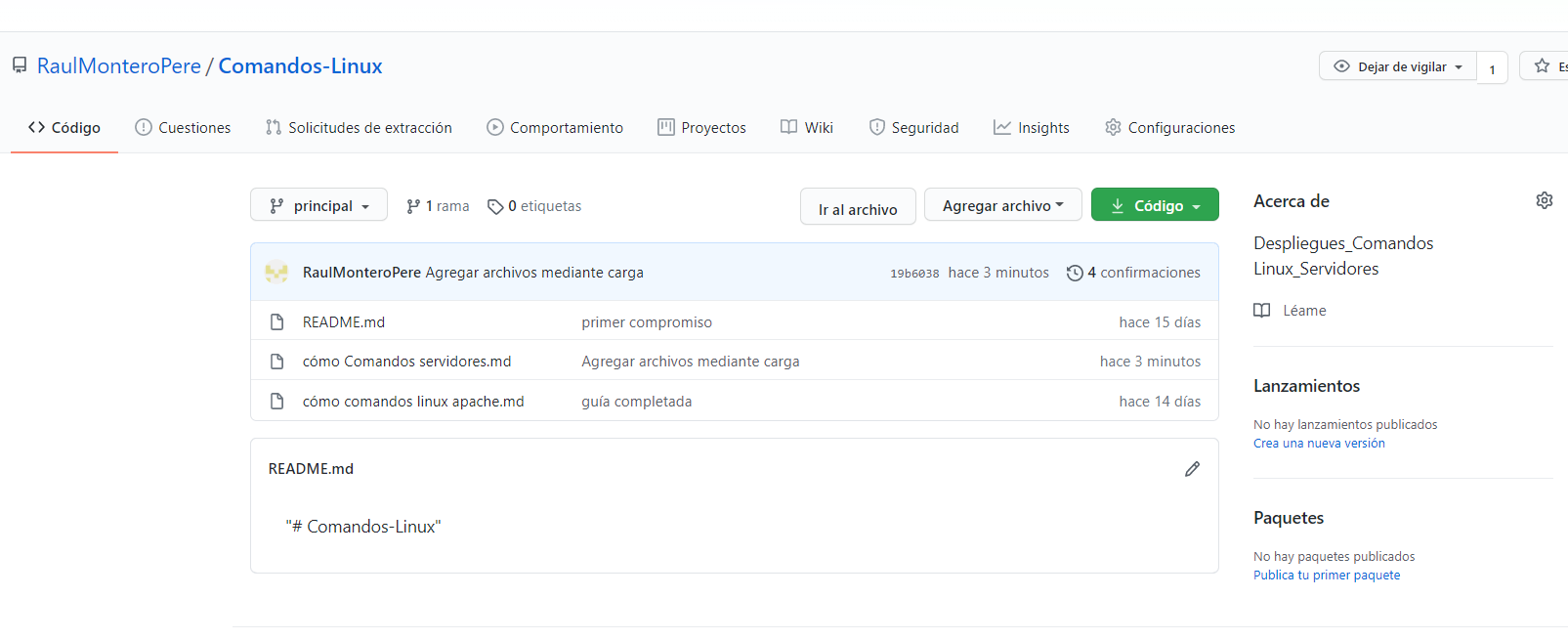
# 4.- Guía “How To” y subida a GitHUB

Con el programa MarkDown elaboraremos una pequeña guía HOW TO indicando de forma clara y concisa los comandos utilizados y porque responden a las preguntas solicitadas.

-------------SE ADJUNTA GUÍA EN EL PROYECTO----------------------



Posteriormente subiremos nuestra guía al repositorio GITHUB:



Repositorio GitHub comandos Linux: <https://github.com/RaulMonteroPere/Comandos-Linux>