**ACTIVIDAD 2. TAREA INDIVIDUAL. CONFIGURACIÓN DEL SERVIDOR REQUERIMIENTO 1**

Página1

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

**Java Apache Tomcat openSSH 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**

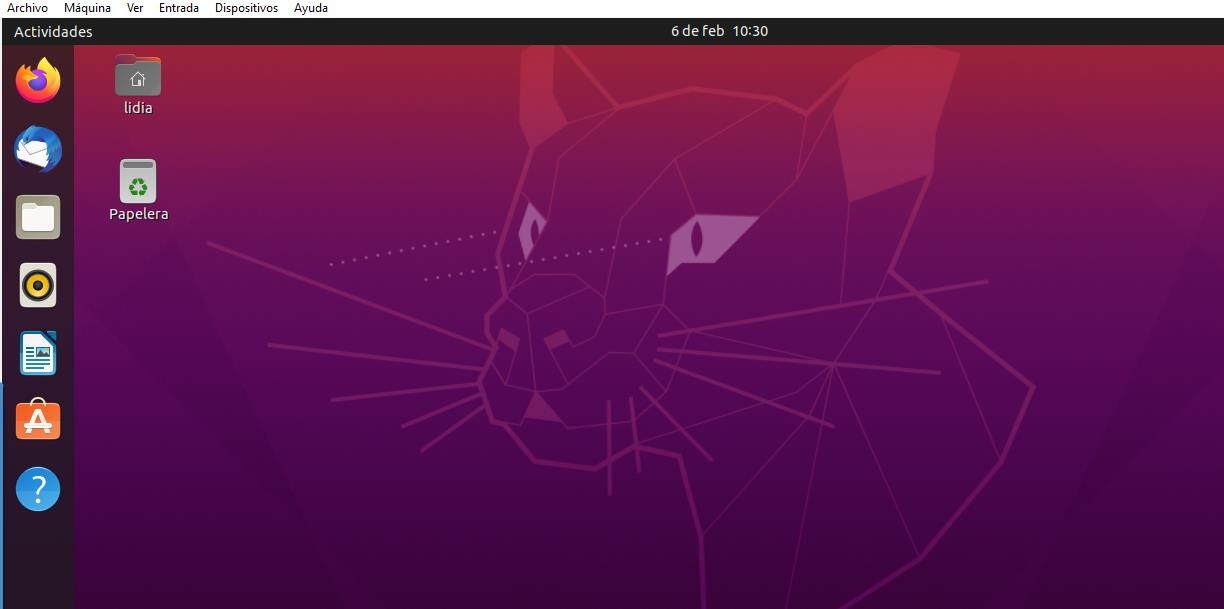
**Para toda la actividad se valorará el orden y la claridad de la documentación, así como la facilidad de uso.**

**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.**

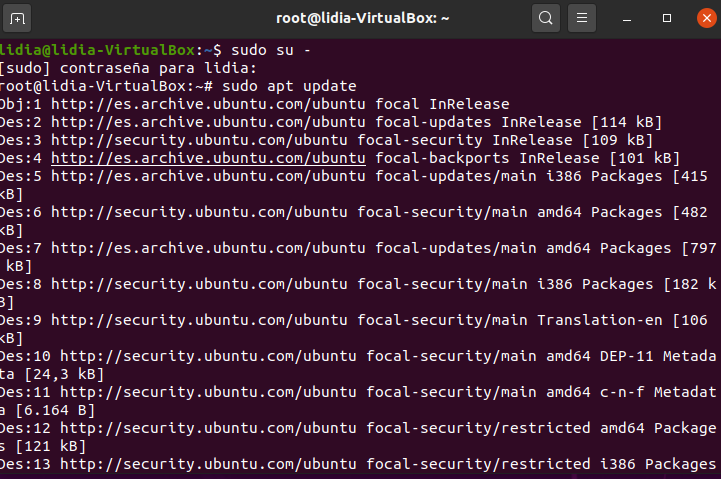
En el primer requerimiento nos solicitan la instalación de una serie de aplicaciones y servidores, por lo que vamos a proceder a su instalación. Para ello seguiremos utilizando la máquina virtual de Ubuntu que nos instalamos en actividades anteriores.

Página2



La primera aplicación que vamos a instalar es Java, por lo que abrimos el terminal. En primer lugar, nos logamos como root para poder realizar la instalación correctamente.

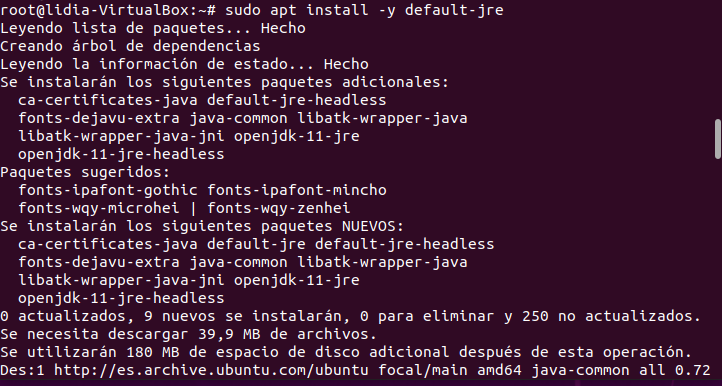
Seguidamente comenzaremos actualizando las listas de paquetes disponibles:



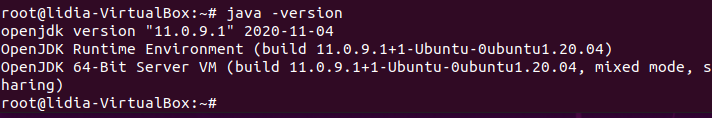
En Ubuntu 20.04 disponemos de las versiones 8, 11, 13 y 14 de Java OpenJDK, correspondiendo la versión por defecto (los paquetes default-jdk y default-jre) a Java OpenJDK 11. En el caso que nos ocupa hemos elegido el JRE.

Por lo que instalamos el paquete con apt.

Página3

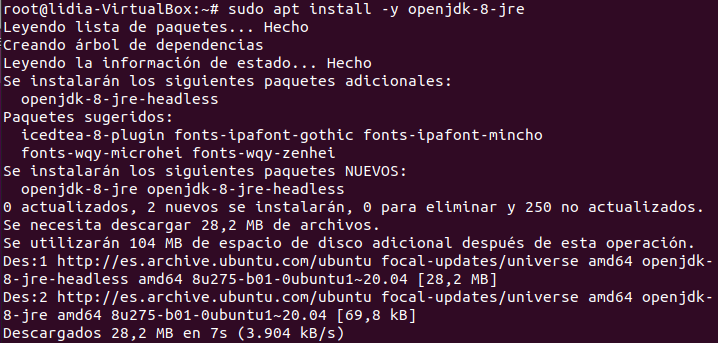


Para comprobar la versión instalada basta con utilizar el comando java -version.



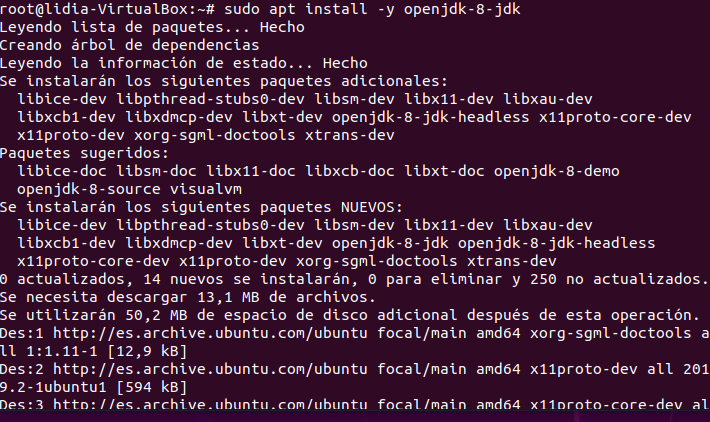
Por otra parte, debemos saber también que en ciertos entornos se necesita la presencia del JRE de Java 8, por lo que será necesario instalar Java OpenJDK 8 en Ubuntu 20.04.

En el caso del entorno de ejecución o JRE de OpenJDK 8 para Ubuntu 20.04 instalamos el paquete openjdk-8-jre:

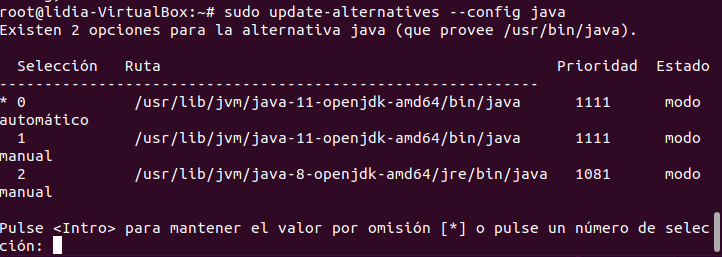


Y para el kit de desarrollo JDK de OpenJDK 8 seleccionaríamos el paquete openjdk-8-jdk:

Página4



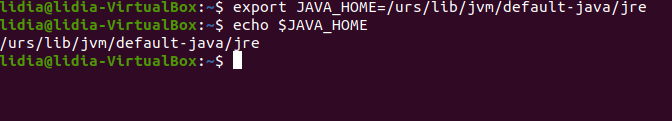
Para establecer la versión de Java que se usará por defecto en Ubuntu 20.04 utilizamos el comando update-alternatives:



La que está marcada con un asterisco es la versión actualmente seleccionada.

Finalmente, como apunte, muchas aplicaciones basadas en Java necesitan conocer la ruta a la instalación de Java en el sistema a través de la variable de entorno JAVA\_HOME.

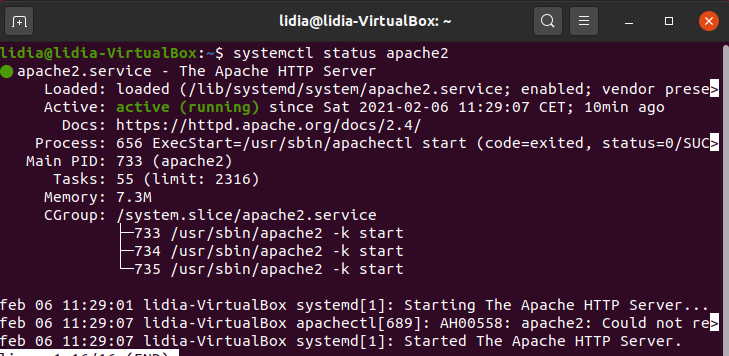
La forma más sencilla de que la variable JAVA\_HOME esté disponible para cualquier usuario del sistema es configurando su valor en /etc/environment:



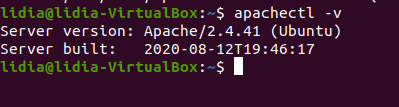
Con estos pasos ya tendríamos instalado y configurada la variable de JAVA\_HOME en nuestro equipo. A continuación, vamos a proceder a instalar Apache.

Como en la actividad anterior ya instalamos el servidor Apache, vamos a comprobar que lo tenemos instalado.

Página5

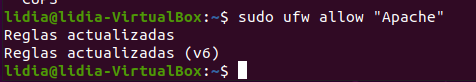


Para averiguar la versión de Apache que tenemos instalada en Ubuntu 20.04 podemos usar el comando apachectl -v.



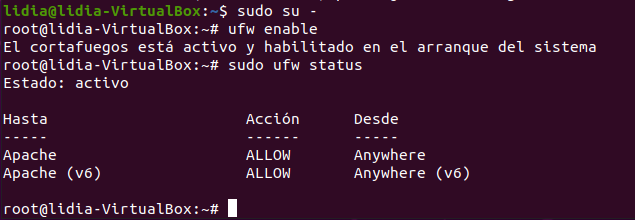
Como ya permitimos el acceso al servidor por SSH debemos habilitar las conexiones con la aplicación Apache. Esta aplicación nos permite el acceso por HTTP por los puertos comunes de HTTP (80) y HTTPS (443).

Esto lo conseguimos con el comando siguiente:

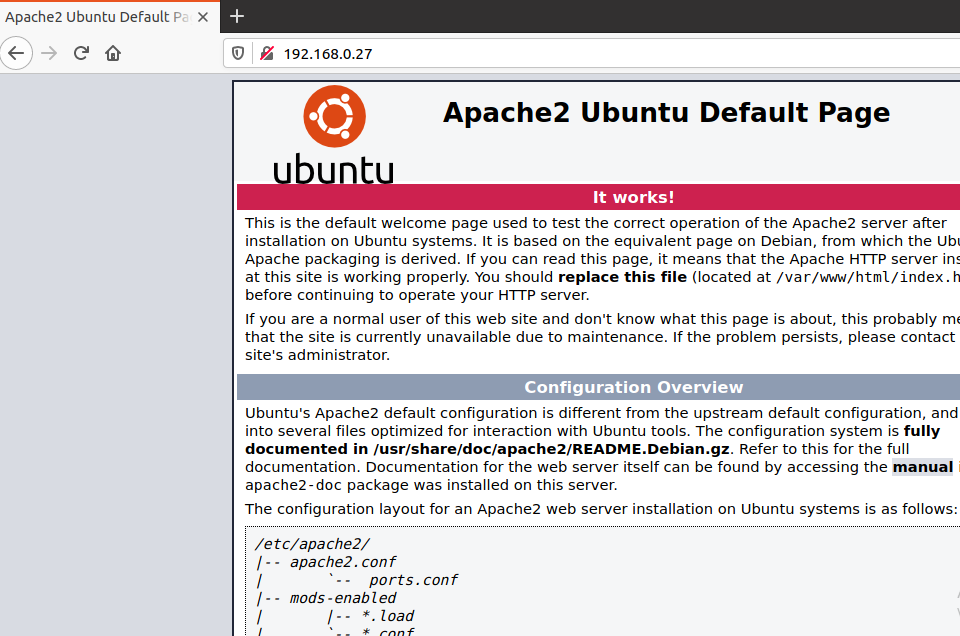


Finalmente, una vez realizada la configuración, podemos activar ya el firewall, con el **comando ufw enable**, también podemos comprobar la configuración de UFW con el **comando ufw status**. Ahora deberíamos ver que el firewall se encuentra activo y además podremos obtener un listado de aplicaciones habilitadas.

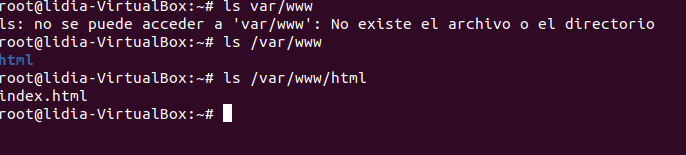
Página6



Lo último que necesitaríamos hacer seria probar el servidor web, para ello debemos saber nuestra IP. Si todo sale bien, como podemos comprobar en la imagen posterior es que ya tenemos instalado nuestro servidor.

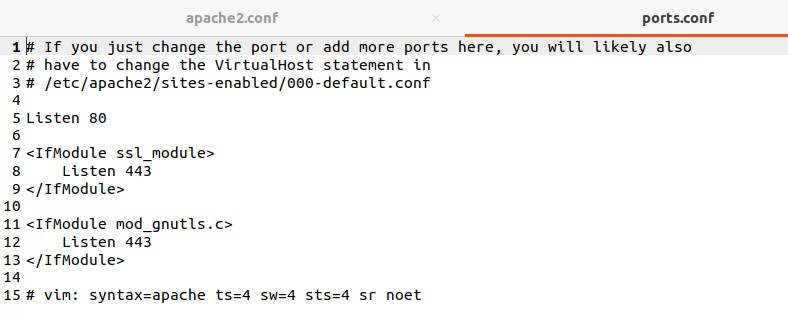


Por otra parte, para saber donde están alojadas las paginas web se usa el comando ls /var/www

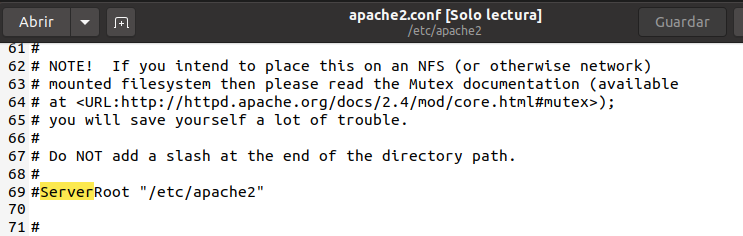


Si nos vamos a los archivos de configuración, vemos que Apache está escuchando por el puerto 80.

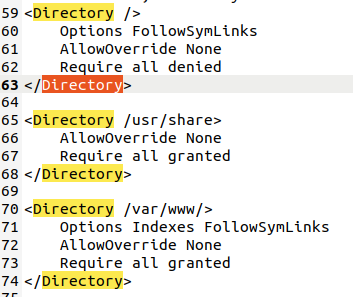
Página7



Y el directorio del ServerRoot

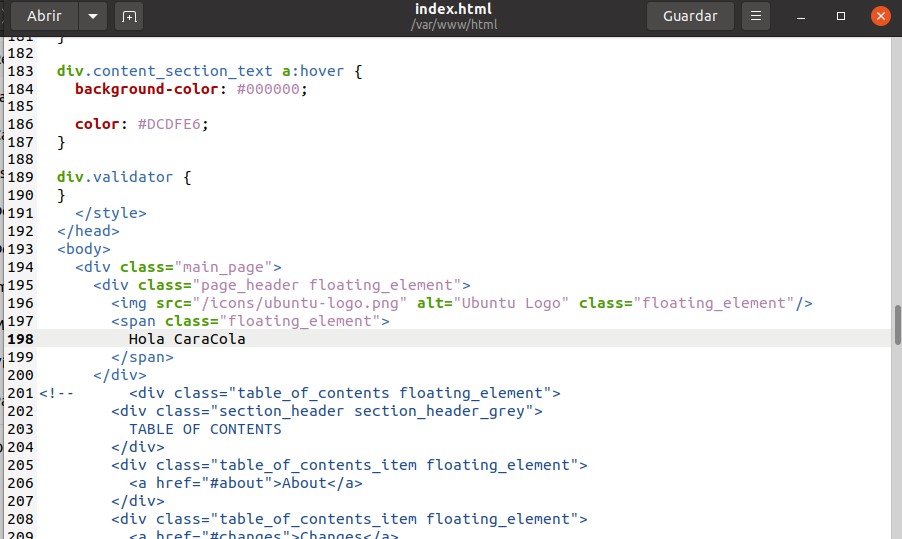


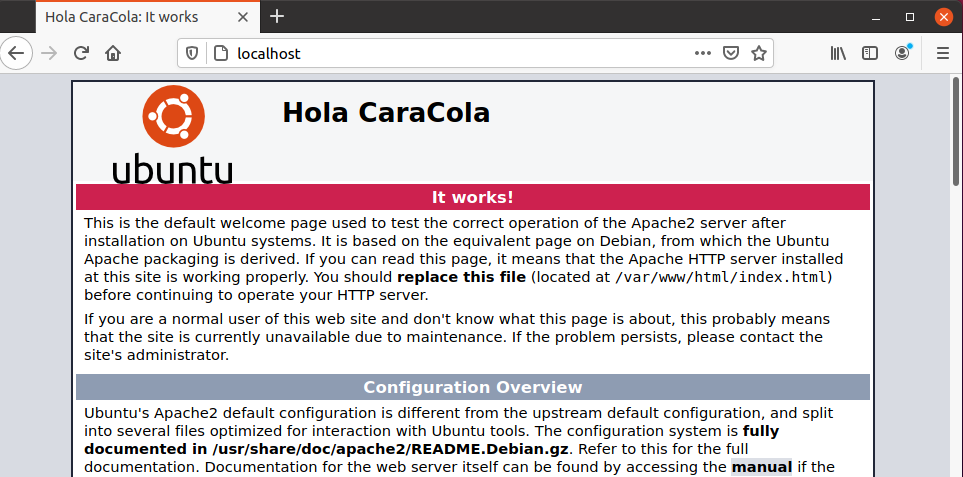
También podemos ver que están denegados los permisos para el acceso a mis documentos (mi directorio raíz), pero sí está permitido el acceso para las páginas webs.



Como tenemos permisos para modificar el archivo HTML, lo modificamos y vemos como efectivamente la página de inicio cambia.

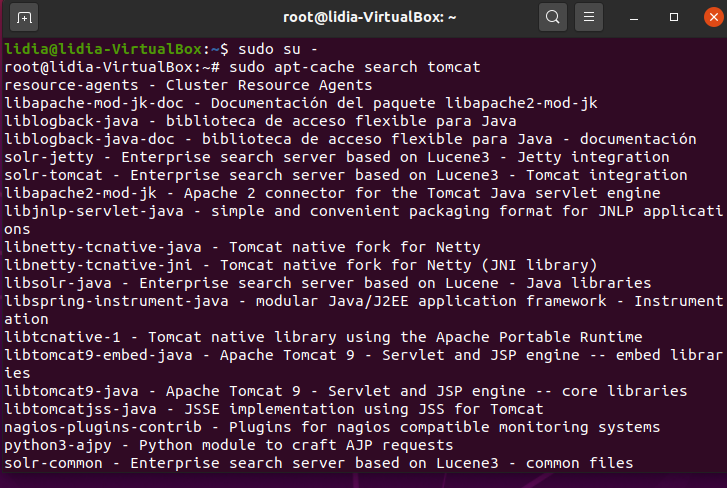
Página8



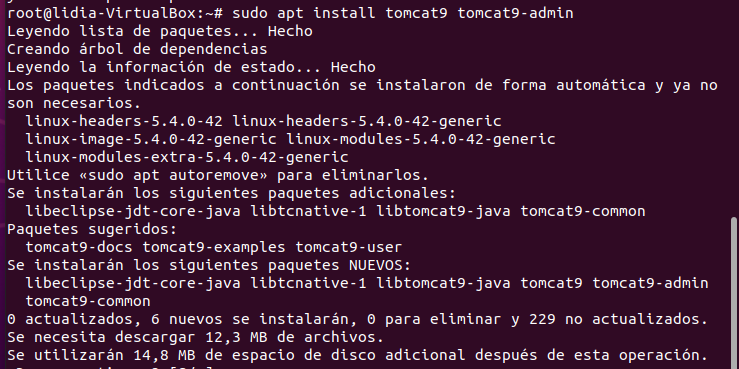


El siguiente servidor a instalar es Tomcat. Lo primero que podemos hacer es consultar el repositorio para ver qué paquete de Tomcat está disponible para descargar.

Página9

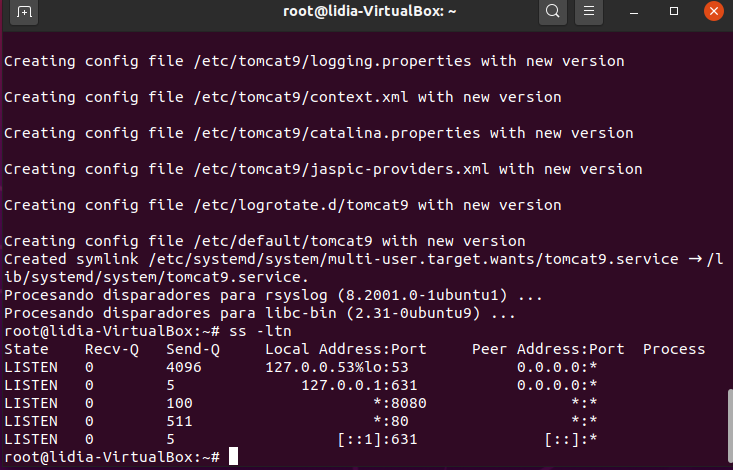


Podemos ver que se nos va a instalar el Apache Tomcat 9. Por lo tanto, vamos a proceder a su instalación. El comando para ello será:

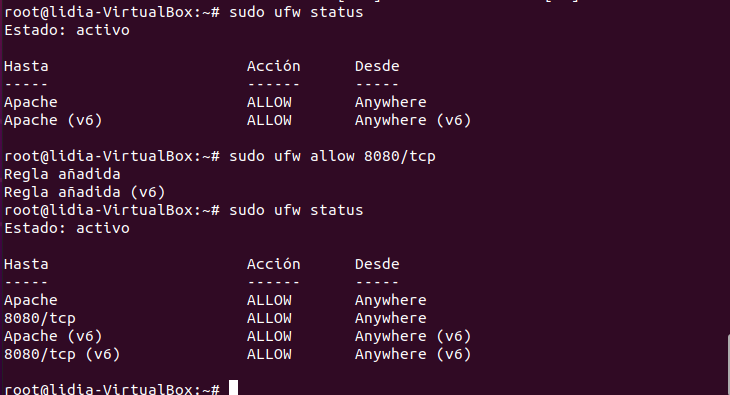


Una vez que Tomcat haya finalizado la instalación, debería iniciarse automáticamente. Se puede verificar que se ejecute con el sscomando. Debería ver un puerto abierto, número 8080, ya que ese es el puerto predeterminado para Apache Tomcat.

Página10

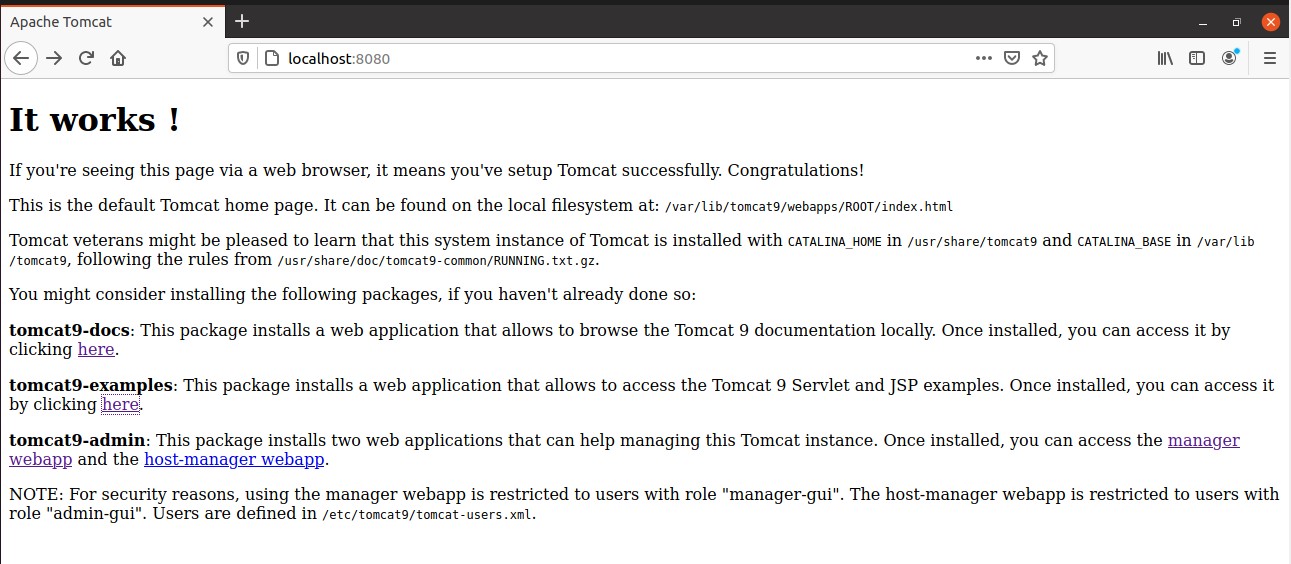


Por otra parte, también necesitamos configurar el firewall para permitir el tráfico desde cualquier fuente al puerto 8080.



Con Tomcat en funcionamiento, ahora deberíamos poder acceder a él en un navegador web: localhost:8080.

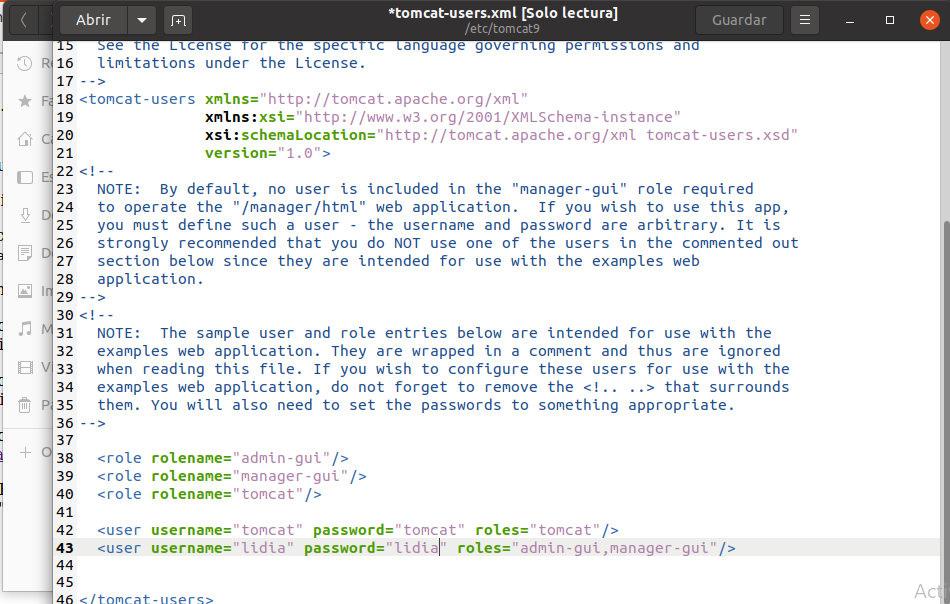
Página11



A continuación, vamos a asignar un usuario para el Administrador de aplicaciones web en el servidor Tomcat. Es por ello que debemos crear una nueva cuenta de usuario para utilizar Apache Tomcat Web Application Manager.

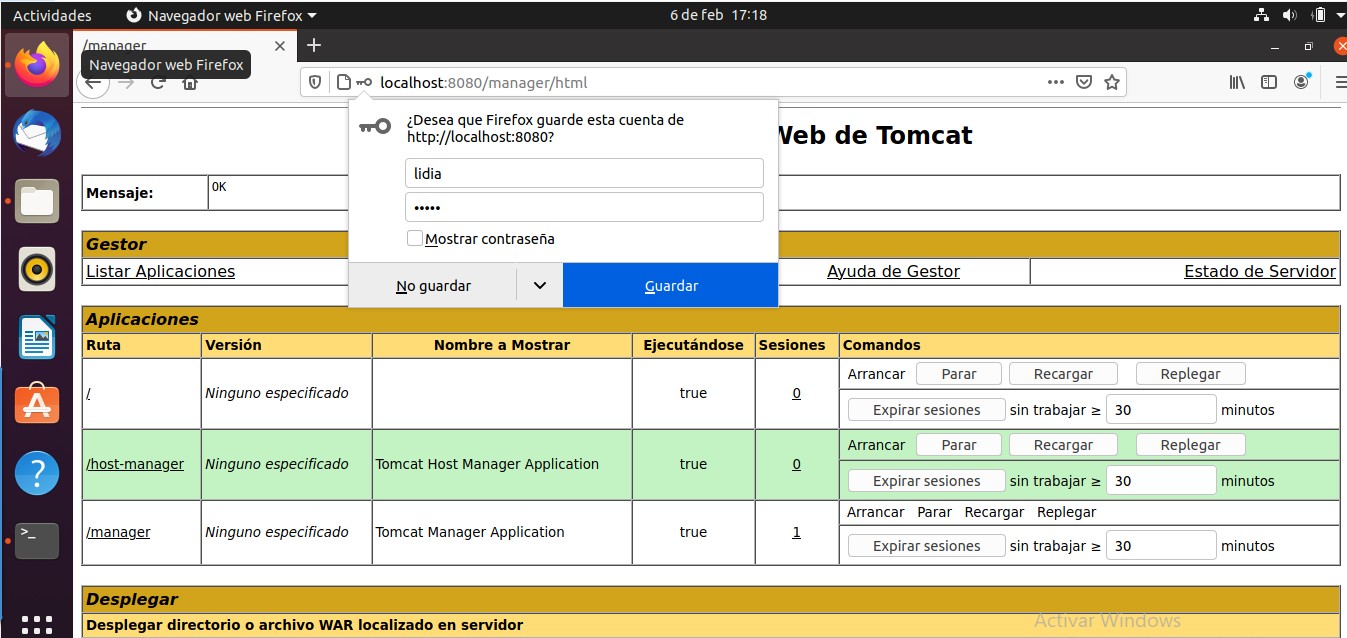
Por tanto, los pasos que seguiremos serán los siguientes. Vamos a abrir el archivo tomcat-users.xml en el directorio Tomcat con el editor de texto.

Cambiamos contraseñas y usuarios junto con los roles de admin y manager como se puede observar en la imagen.



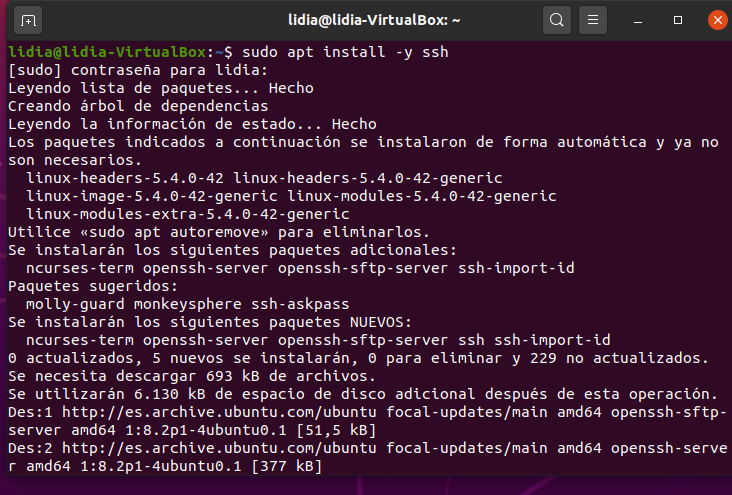
Seguidamente y después de reiniciar el servicio, deberemos iniciar sesión en Tomcat Web Application Manager y nos solicitará las credenciales que acabamos de asignar.

Página12

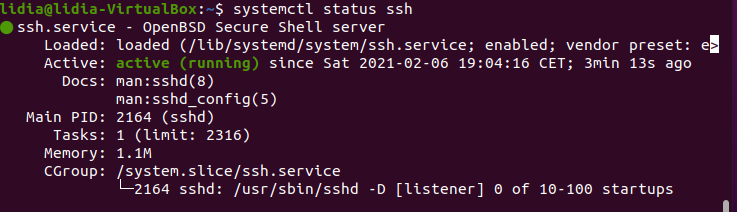


Con este paso ya tendríamos instalado correctamente nuestro servidor Tomcat.

La siguiente tarea que se nos solicita es instalar OpenSSH. Instalamos este paquete con apt.



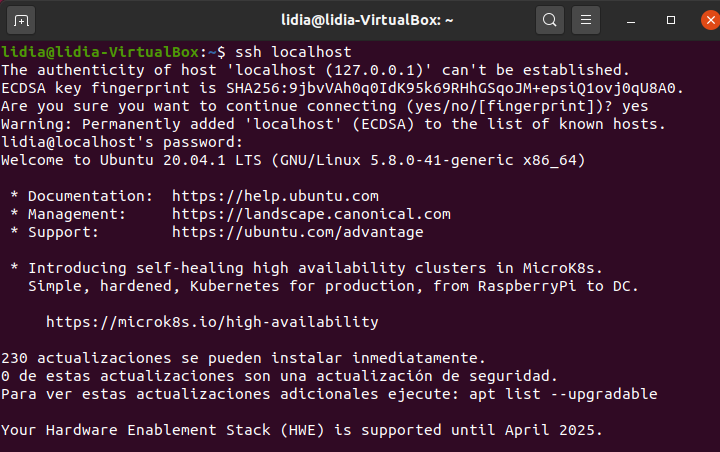
Podemos comprobar el estado del servicio SSH.



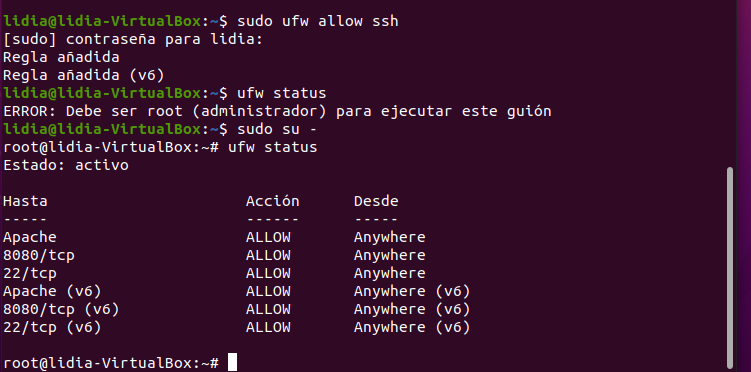
Además, podemos comprobar el servicio SSH en modo local, usando el comando de consola ssh localhost:

Página13

La primera vez que conectamos con un cliente SSH a un servicio se nos pregunta si deseamos guardar la huella digital de la clave pública que usa el servidor, para que en próximas conexiones se compare la huella guardada con la recibida en la nueva conexión, puesto que si son distintas indicaría que los certificados ya no son los mismos, siendo una posible causa la suplantación de identidad del sitio.



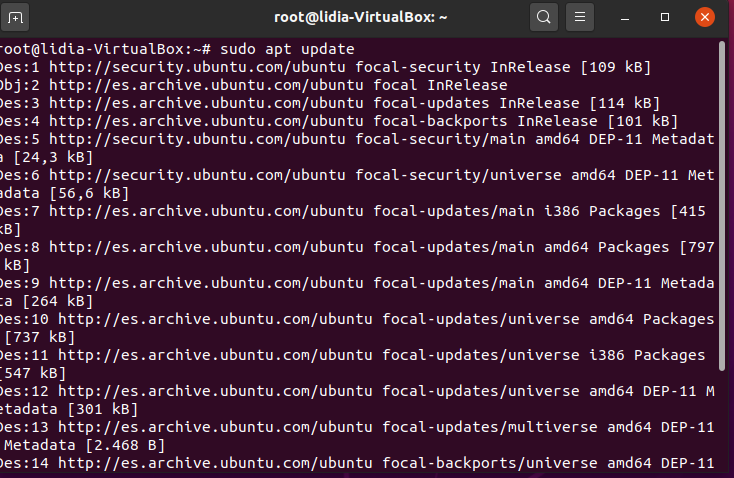
Finalmente, necesitamos configurar el firewall para realizar conexiones desde red al servicio SSH.



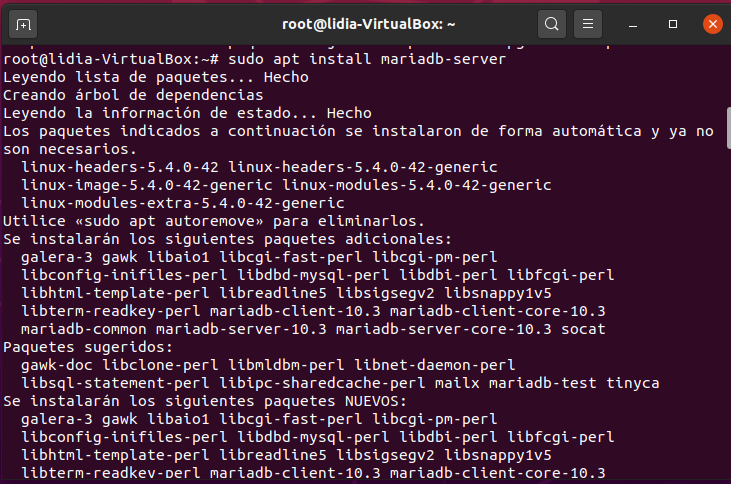
El último paso de esta actividad es instalar MariaDB, un sistema de administración relacional de bases de datos de código abierto.

Inicialmente lo que necesitamos hacer es actualizar el índice de paquetes usando apt.

Página14

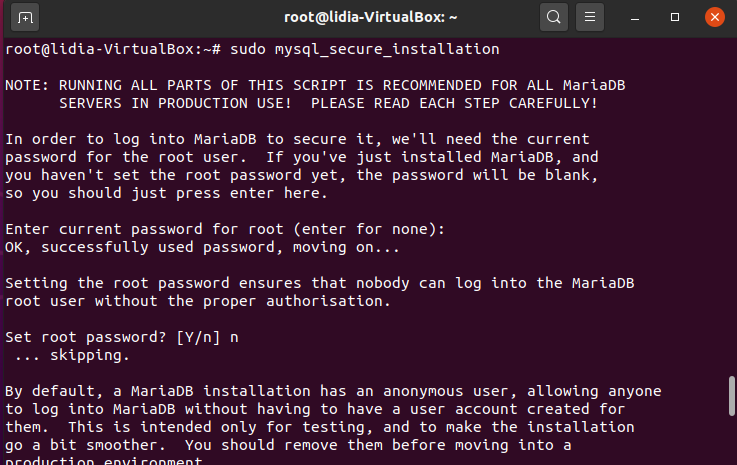


Seguidamente, instalamos el paquete de mariadb-server usando apt.



En las nuevas instalaciones de MariaDB, el siguiente paso es ejecutar la secuencia de comandos de seguridad incluida. Esta secuencia de comandos cambia algunas de las opciones predeterminadas que son menos seguras. La usaremos para bloquear las conexiones de root remotas y eliminar los usuarios de la base de datos no utilizados.

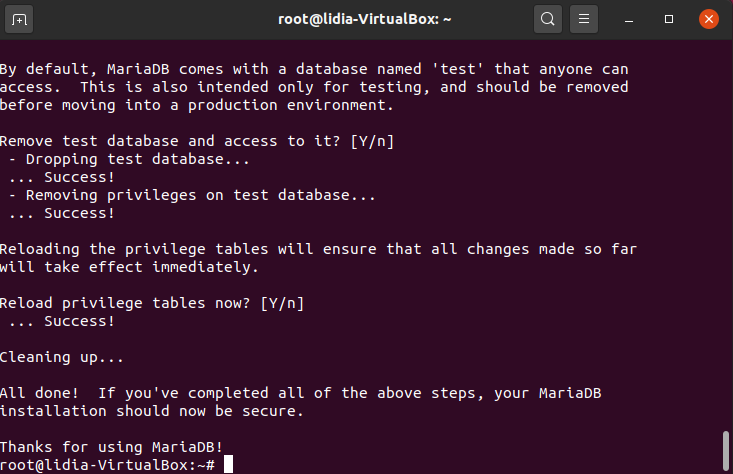
Página15



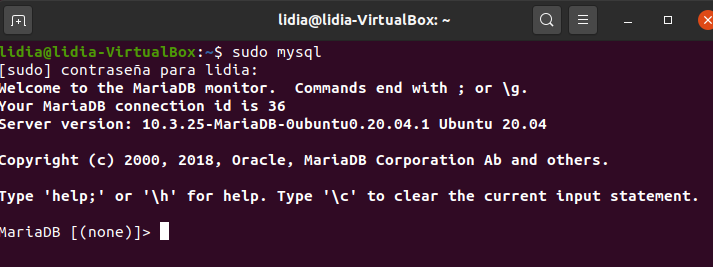
Ahora, veremos una serie de solicitudes mediante las cuales podremos realizar cambios en las opciones de seguridad de su instalación de MariaDB. En la primera solicitud se pide que metamos la contraseña root de la base de datos actual. Debido a que no configuramos una aún, pulsamos no y le damos a enter.

Para todas las siguientes cuestiones, también pulsamos enter y de este modo aceptamos los valores predeterminados para todas las preguntas siguientes. Con esto, se eliminarán algunos usuarios anónimos y la base de datos de prueba.

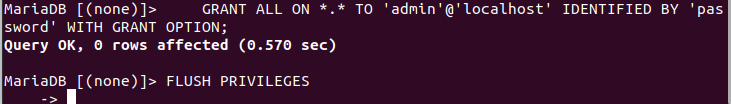
Página16



El siguiente paso será crear una cuenta nueva llamada administrador con las mismas capacidades que la cuenta root, ya que normalmente se recomienda crear una cuenta administrativa independiente para el acceso basado en contraseña. Para hacer este paso introducimos en el terminal el siguiente comando:

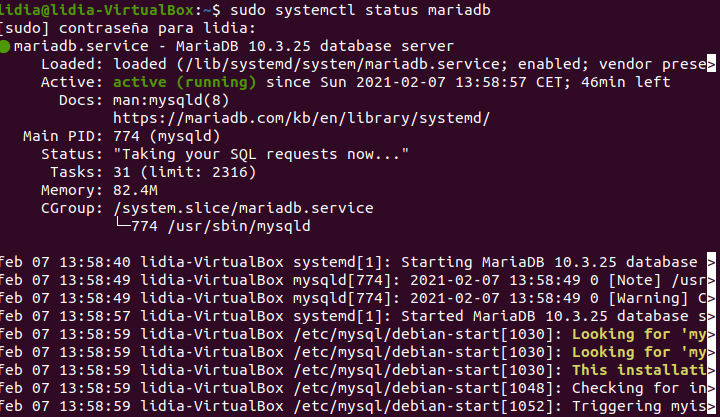


Ahora, crearemos un nuevo usuario con privilegios root y acceso basado en contraseña y vaciamos los privilegios.



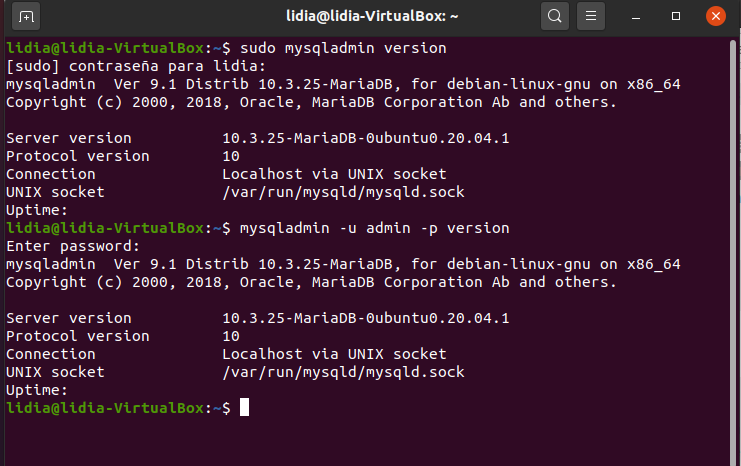
Seguidamente cerramos el terminal y pasamos a probar la instalación.

Página17



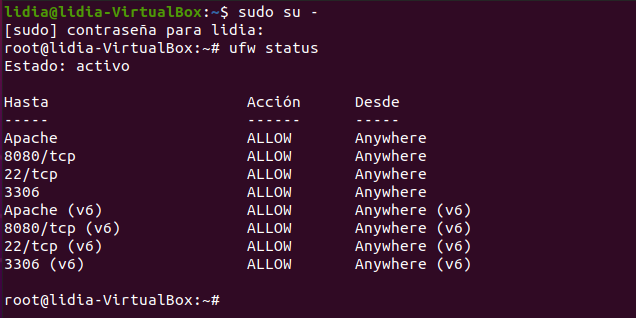
Como comprobación adicional, puede intentar establecer conexión con la base de datos usando la herramienta mysqladmin, que es un cliente que le permite ejecutar comandos administrativos.

Además, como hemos configurado un usuario administrativo independiente con la autenticación de contraseña, podemos realizar la misma operación tal y como se muestra en la imagen posterior.



Esto significa que MariaDB está activo y que nuestro usuario puede autenticarse correctamente.

Página18



Con esta última instalación damos por terminada la segunda actividad.