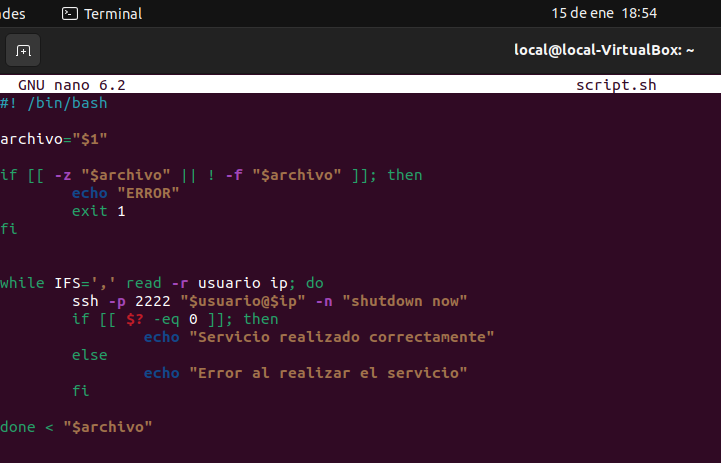
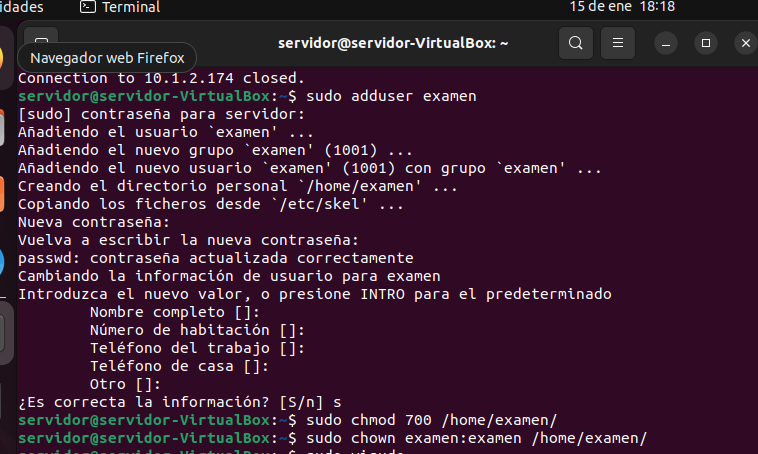
1. **Crea un script para bash que realice el apagado de todos los equipos de nuestra red. Para poder llevar a cabo, este recibe como parámetro de entrada el nombre de un fichero de datos con la siguiente estructura:**

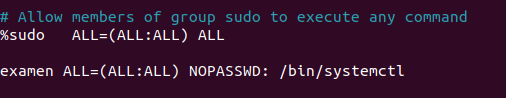


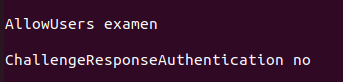


Como salida, se informará por pantalla de la evolución de la ejecución del script

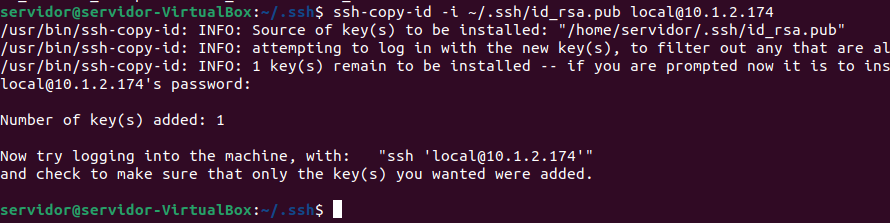
* Crearemos un nuevo usuario para ejecutar la tarea



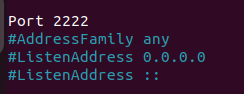




* El acceso a la máquina solamente se podrá realizar mediante claves asimétricas, quedando deshabilitado cualquier otro tipo de acceso



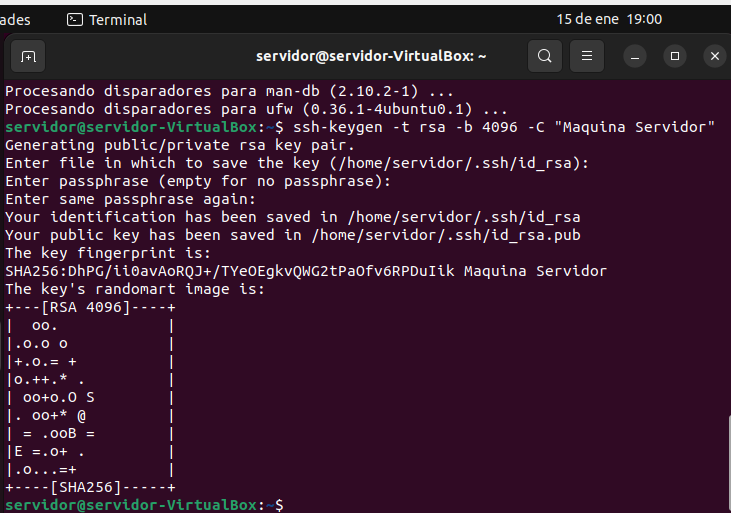
* El acceso remoto como usuario root, vía SSH, debe estar deshabilitado
* El puerto de acceso no podrá ser el 22



* La utilización de sudo estará restringido a determinados comandos
* Solo aceptara peticiones de conexión desde una máquina determinada

1. **Detalla cómo crear una clave pública/privada con frase de paso. Una vez hecho, explica cómo configurar el sistema para utilizar el SSH agent forwarding y SSH X11**

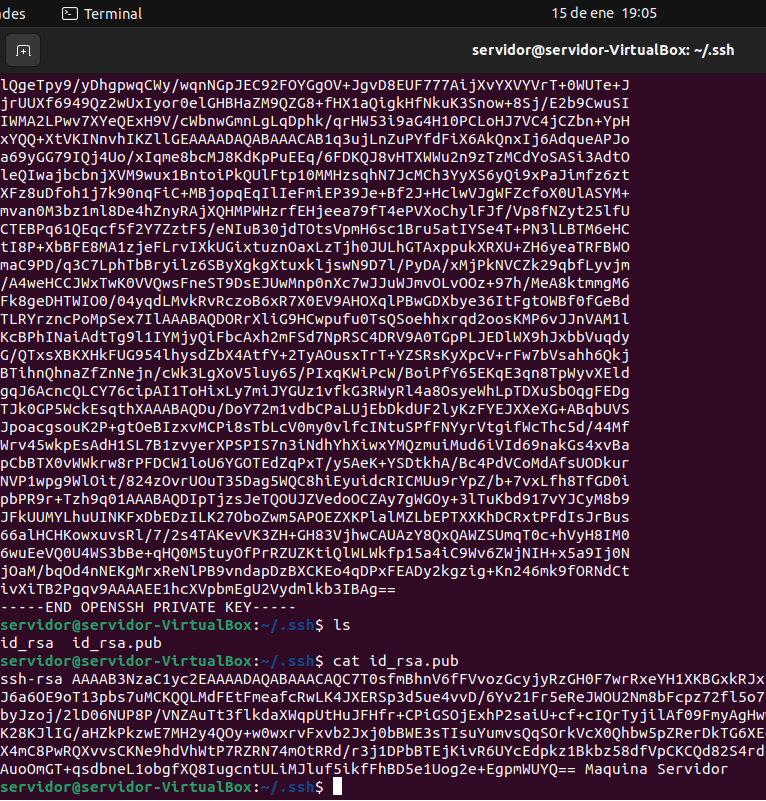
Para crear una clave pública/privada necesitaremos usar el siguiente comando:



La diferencia con la clave sin frase es que en este caso añadimos “-b 4096 -C”, este número hace referencia por el puerto que usará para la autenticación de la clave y el -C se usa para poder añadir la frase de paso.

Para comprobar esta clave entraremos en el directorio ssh, en el podremos listar las dos claves:

* id\_rsa → Clave privada
* id\_rsa.pub → Clave publica

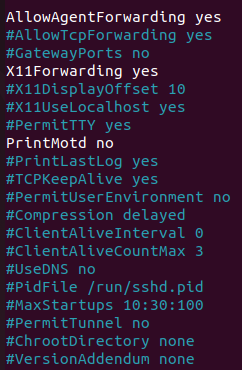


Como podemos ver la clave pública contiene el mensaje que hemos añadido al crear la clave pública/privada.

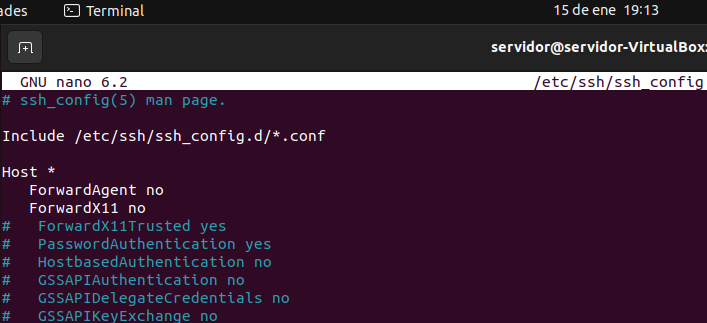
**Configurar el sistema para usar SSH Agent Forwarding y SSH X11:**

La configuración de ellos se encuentra en el mismo archivo de edición: /etc/ssh/sshd\_config

Para habilitarlos solo bastaría con desmarcas las líneas comentadas correspondientes:



Una vez descomentados deberemos editar el archivo de configuración: /etc/ssh/ssh\_config



Por último reiniciamos los servicios de sshd y ssh para aplicar la configuración.

1. **Explica que es un túnel en modo local port forwarding, como crearlo y cómo utilizarlo. Además deberemos hacer lo mismo para un túnel en modo remote**

* Local Port Forwarding:

**Explicación**

Este tipo de túnel permite redirigir un puerto local en la máquina cliente (ya sea local o destino) a un puerto específico en un host. Esto permite acceder al servicio viendo como si fuese localmente.

**Creación**

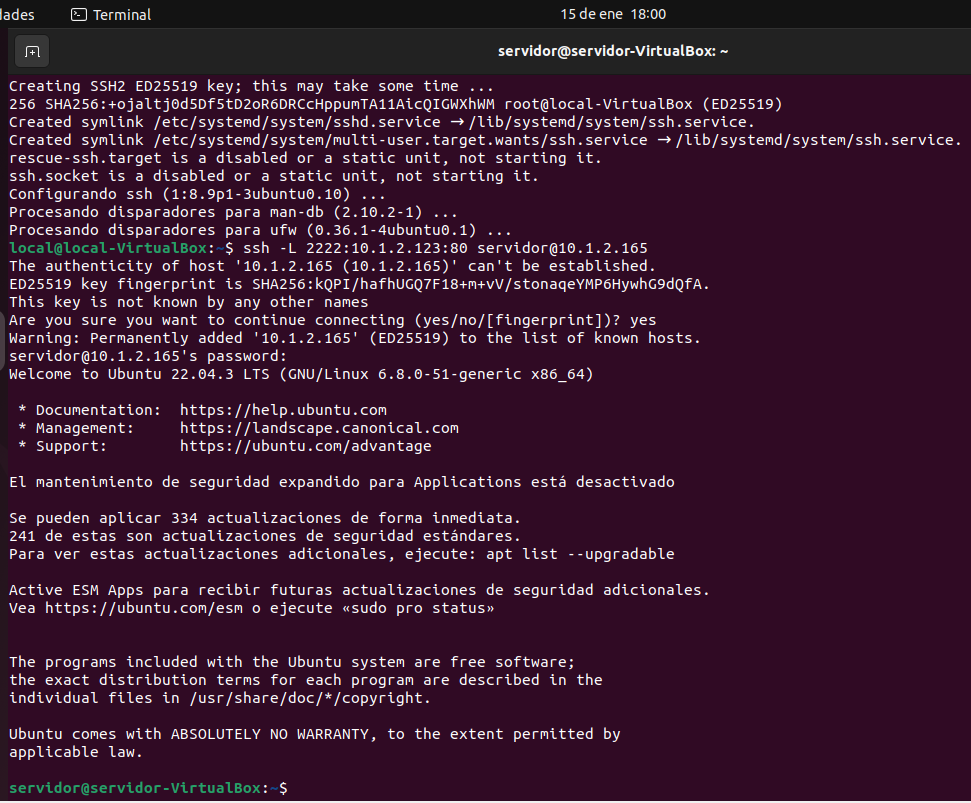
Para la creación de este túnel vamos a necesitar 3 maquinas

* Local → Máquina la cual quiere acceder al servicio (Apache)
* Servidor → Actuará como puente entre las dos máquinas
* Destino → Maquina que contiene el servicio Apache

Para la creación del túnel necesitaremos instalar en el servidor openssh-server y en las dos máquinas clientes openssh-client. Además en la máquina llamada Destino instalaremos el servicio Apache2

Una vez tenemos todo instalado aplicaremos desde VirtualBox la opción de Adaptador puente en la Red. Esto nos ayudará para conseguir que se vean las máquinas.

Una vez aplicado, entraremos en la máquina Local para usar el siguiente comando:



Este indica que nos conectaremos a la máquina Destino (10.1.2.123) a través del puerto 2222 accediendo mediante el servidor que actúa como puente.

**Utilización**

Para comprobar la funcionalidad de este puente entramos en Firefox desde la máquina Local y usando la dirección de localhost con el puerto asignado, comprobaremos:



* Remote Port Forwarding:

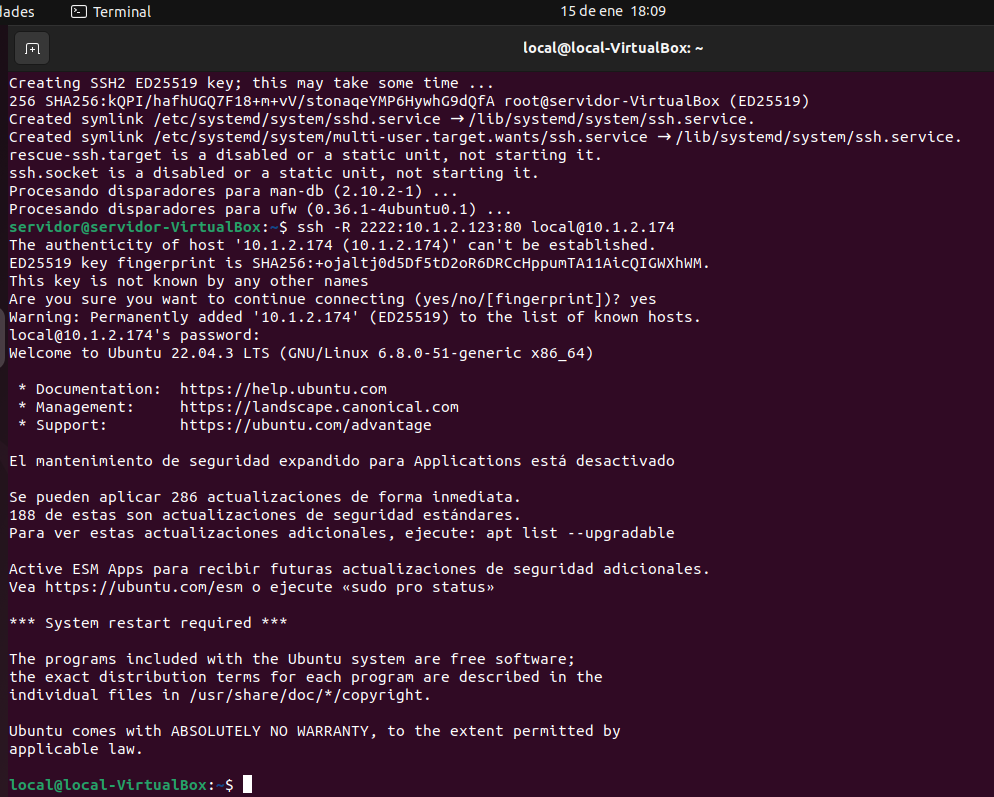
**Explicación**

Permite que un puerto en localhost sea redirigido a la máquina cliente sin necesidad de usar el túnel. En pocas palabras la máquina Destino usa al servidor como puerto para que otra máquina (local en este caso), pueda acceder a el servicio de Apache

**Creación**

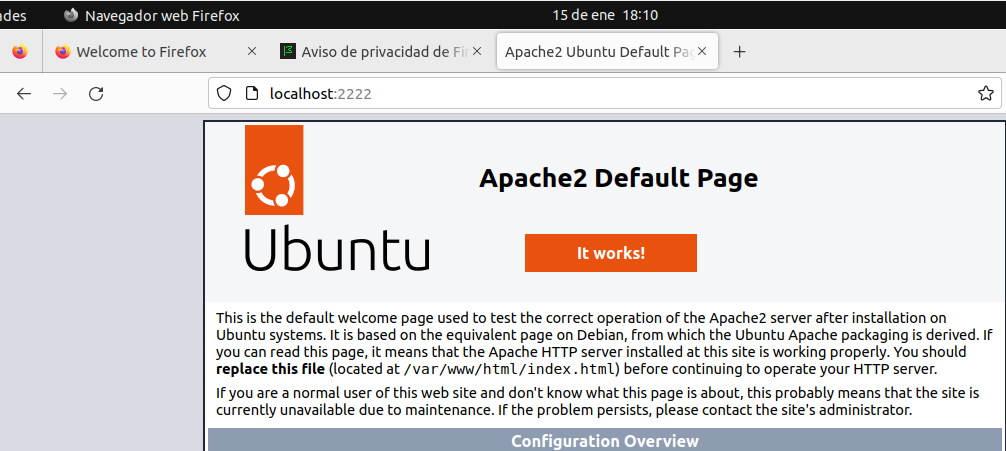
* Local → Máquina la cual quiere acceder al servicio (Apache)
* Servidor → Será el encargado de transmitir a otras máquinas el servicio
* Destino → Maquina que traspasa el servicio al servidor

En este caso crearemos la conexión desde la máquina servidor, usando el comando:



**Utilización**

Y por último deberemos ver desde la máquina local el servidor de apache usando la ruta de dirección del servidor. En este caso vemos el servidor de apache correctamente



1. **¿Qué es PAM? Asociado a PAM, explica que directorios y ficheros tienen gran importancia. Ademas de lo anterior, detalla como deshabilitar el acceso via SSH de cualquier usuario que no sea root y cambia el mensaje del dia cuando un usuario se conecte via SSH**

PAM es un mecanismo de autenticación flexible que permite a las aplicaciones y otro software del proceso de identificación.

1. Contiene un fichero de configuración para cada programa que use PAM:
   * /etc/pam.d/login
   * /etc/pam.d/sshd
   * /etc/pam.d/lightdm
   * /etc/pam.d/passwd
2. **Fichero por defecto**: /etc/pam.conf  
   Define los módulos PAM utilizados por los programas que emplean PAM, como login, sshd y passwd.

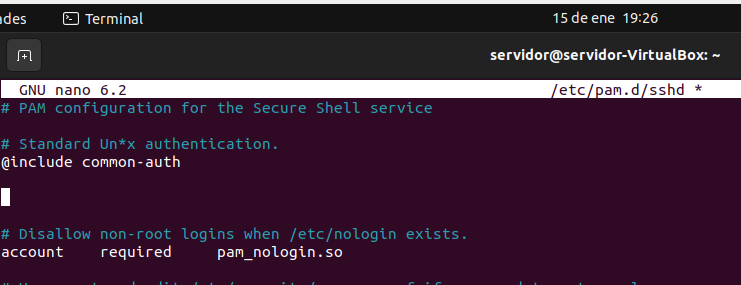
Por ultimo tenemos el archivo de configuracion /etc/pam.d/su que contiene los permisos de configuracion: required, requisite, optional y sufficient. Si la configuracion de required no se encuentra el programa no podra continuar

**Deshabilitar el acceso vía SSH:**

Para deshabilitar el acceso deberemos editar el archivo de configuración de PAM: /etc/pam.d/sshd

Una vez aquí añadiremos la línea de configuración que nos permite bloquear el acceso en el usuario:

auth required pam\_listfile.so item\_user sense=deny file=/etc/pam.d/allowed\_users



Una vez añadida esta línea deberemos crear un archivo con este mismo nombre (allowed\_users) donde permitiremos al usuario que queramos, por ejemplo guti-server. Este sería el único usuario que podrá acceder vía SSH.

Además necesitaremos otro archivo donde se rechazará el mensaje en caso de que el usuario que quiera entrar no sea el aceptado por PAM

**Cambiar el mensaje del dia:**

Para cambiar el mensaje