**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

**Administración de Servicios en Red**

**Profesor:**

**Soto Ramos Manuel Alejandro**

**Alumnos:**

**Manuel Jesús Arreola Sandoval**

**Miguel Ángel Jiménez Ríos**

**Carlos Aníbal Larios Moguel**

**Grupo:**

**4CV2**

**Primer Parcial Semestre 2018-2**

**Práctica # 6**

**Instalación y configuración OpenSSH.**

[**1. Instalación.**](#_gjdgxs) **3**

[**2. Configuración.**](#_30j0zll) **5**

[2.1 Configuración Inicial.](#_1fob9te) 7

[2.2 Configuración de Login](#_3znysh7) 8

[2.3 Límite de Intentos Fallidos por Tiempo y por mala contraseña.](#_2et92p0) 10

[2.4 Conexión Segura utilizando llaves de Identificación.](#_tyjcwt) 10

[2.5 Configuración de Logs.](#_3dy6vkm) 11

[2.6 Ejecución de Aplicaciones de GUI de forma Remota.](#_1t3h5sf) 12

[2.7 Copiar y Administrar Recursos del Sistema de Archivos de forma Remota.](#_4d34og8) 12

[**3. Brindar servicios locales por SSH.**](#_2s8eyo1) **13**

[**4. Pruebas.**](#_17dp8vu) **14**

[4.1 Lista de usuarios y máximo intento de Accesos.](#_3rdcrjn) 16

[4.2 Inicio de sesión por certificado.](#_26in1rg) 17

[4.3 Forwarding de aplicaciones gráficas.](#_lnxbz9) 19

[4.4 Denegación de usuario root.](#_35nkun2) 22

[4.5 Inicio de sesión por contraseña.](#_1ksv4uv) 23

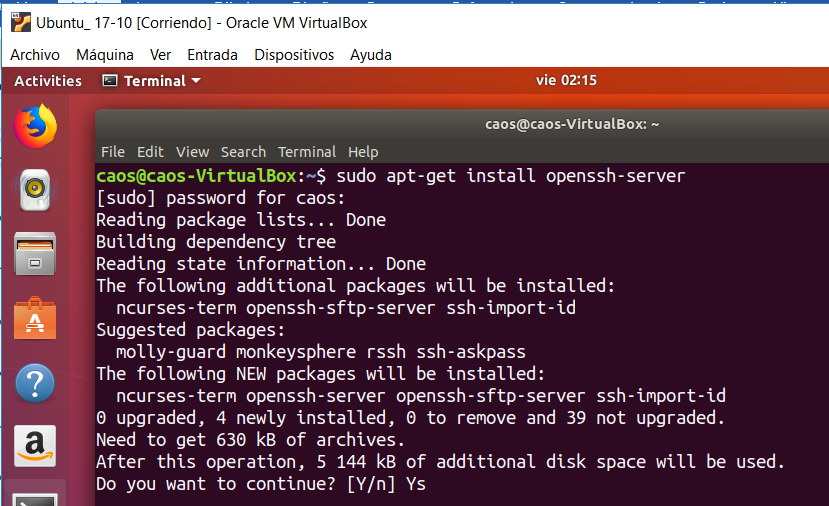
[4.6 Servicios por túnel SSH.](#_44sinio) 23

## 1. Instalación.

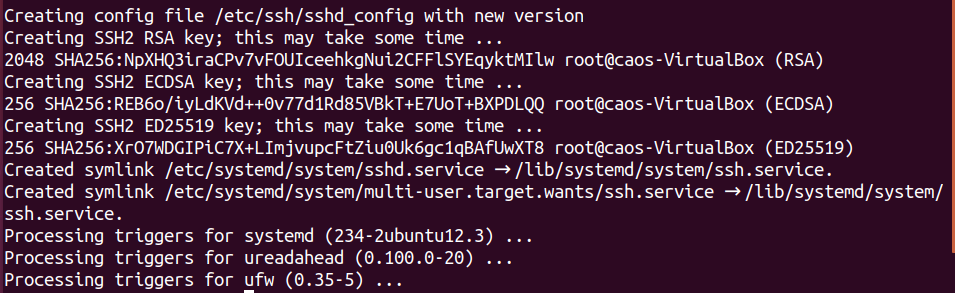
Trabajaremos en la versión 17.10 de Ubuntu a través del Hypervidor VirtualBox desde una laptop Lenovo Y700 con Sistema Operativo Windows 10 Home Edition.

Para instalar los paquetes openssh-server y openssh-client abriremos una ventana de terminal (CTRL + ALT +T) y digitaremos para instalar el Servidor:

$ sudo apt-get install openssh-server

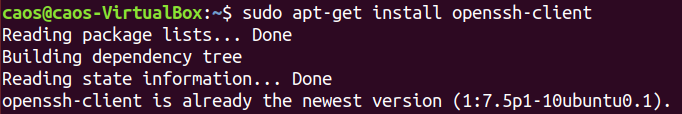


El instalador nos creará los siguientes archivos y llave SSH:



Para instalar el Cliente debemos teclear desde Terminal (es probable que ya se encuentre instalado, ya que varias versiones de Linux basadas en Debian traen el cliente instalado por defecto) lo siguiente:

$ sudo apt-get install openssh-client

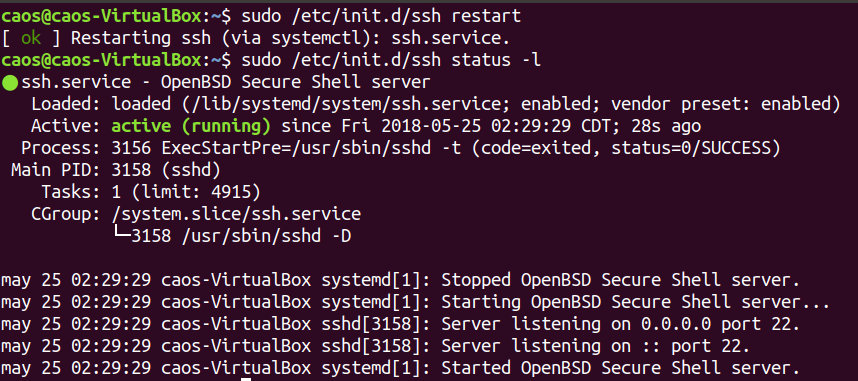


Como root reiniciamos el Servicio y Observamos que se encuentre Activo:

$ sudo /etc/init.d/ssh restart

Al comprobar el estado del servicio se puede observar en qué dirección IP y puerto se encuentra escuchando el servidor:

$ sudo /etc/init.d/ssh status -l

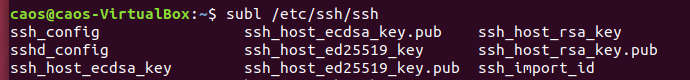


Por defecto, el puerto para el servicio SSH es el 22.

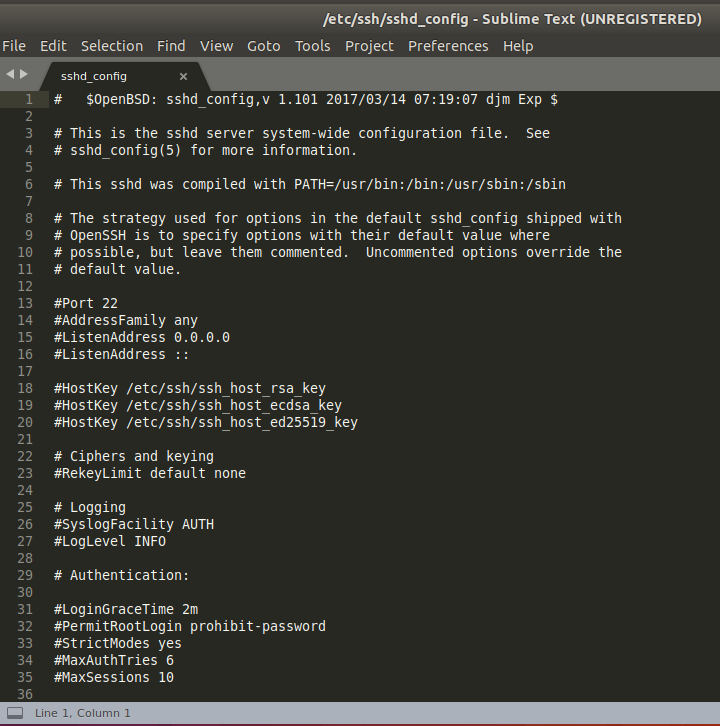
Además podemos observar que escucha en la dirección IP 0.0.0.0; lo cual indica que escuchará en todas las interfaces que tenga.

## 2. Configuración.

La configuración del servidor la realizaremos modificando el archivo /etc/ssh/sshd\_config.

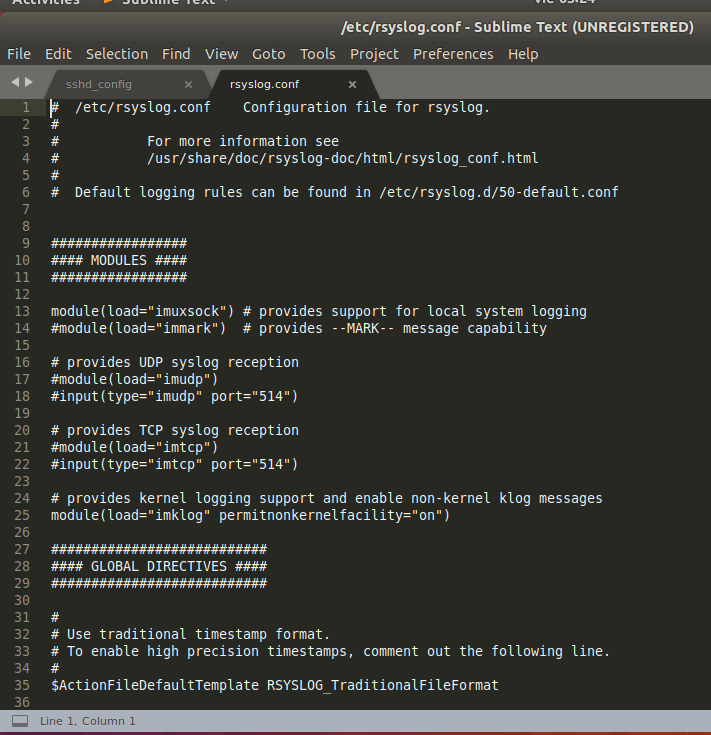


Podremos observar todas las opciones relacionadas a este servidor.



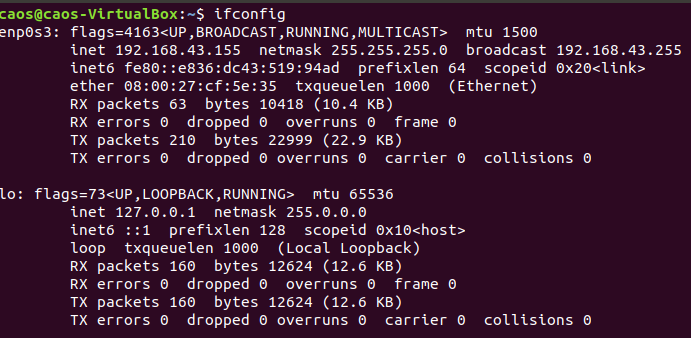
Por último, para activar la escritura de logs en archivo tendremos que modificar el archivo /etc/rsyslog.conf.





### 2.1 Configuración Inicial.

Dejaremos el servicio en el puerto 22 y cambiaremos la dirección de escucha a la IP fija 192.168.43.155, la cual es la que tiene asignada la máquina virtual mediante DHCP ya que corresponderá al equipo que ejecutará el servidor SSH.





### 2.2 Configuración de Login

Para esta configuración tomaremos en cuenta los siguientes puntos a la hora de permitir el acceso de los usuarios del sistema:

1. Denegar el acceso al usuario root.

2. Permitir el acceso sólo a algunos usuarios.

Para el primer punto editamos la siguiente línea:



Para el segundo punto se utiliza la directiva AllowUsers.

Ingresaremos una lista de usuarios separada por espacios.

Cada usuario tendrá la forma user@dirección\_IP, user, dirección\_IP.

De esta manera se puede restringir el login de los usuarios y las direcciones IP desde las cuales pueden acceder al servidor.

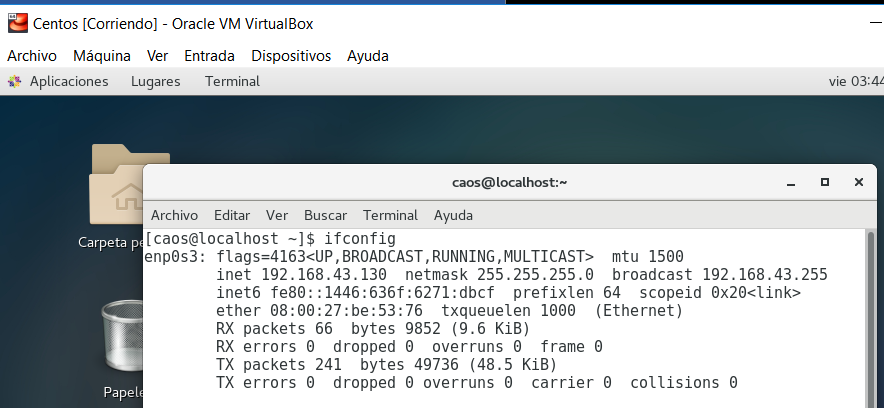
Definiremos 3 usuarios y su acceso será permitido de la siguiente forma:

● “miguel”, en la dirección IP 192.168.43.130

● “anibal”, desde cualquier dirección IP

● “manuel”, desde cualquier dirección IP

El usuario miguel estará desde una Máquina Virtual en Centos 7:

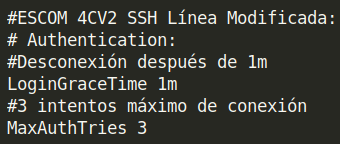


Editamos el archivo de configuración de la siguiente forma:



### 2.3 Límite de Intentos Fallidos por Tiempo y por mala contraseña.

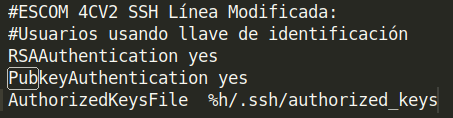
Para configurar el número de intentos válidos y el tiempo máximo para realizar un login sin errores se configuran los parámetros LoginGraceTime y MaxAuthTries:



### 2.4 Conexión Segura utilizando llaves de Identificación.

Si deseamos configurar el acceso al servidor mediante llaves (sin necesidad de hacer login por contraseña), se debe habilitar la opción correspondiente en el servidor y cada cliente que desee utilizar este método debe generar en su equipo su certificado y llave correspondiente.

Para este caso se utilizan llaves RSA. En el servidor se debe permitir la autenticación por medio de llaves RSA e indicar la ruta de las llaves autorizadas para acceder, de la manera siguiente:



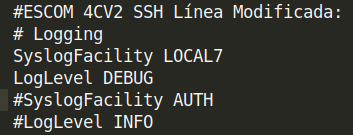
El indicador “%h” referencía el directorio home. De esta forma cada usuario habilitado para iniciar sesión por llave, deberá colocar la llave generada en su equipo, en la respectiva carpeta .ssh/ dentro del servidor ssh.

### 2.5 Configuración de Logs.

Para poder observar y configurar los logs del sistema modificaremos la facilidad del sistema que queramos y el nivel de información que queramos conocer de los logs.

Los niveles que podemos editar son.

QUIET, FATAL, ERROR, INFO, VERBOSE, DEBUG, DEBUG1, DEBUG2 y DEBUG3, el default es INFO.



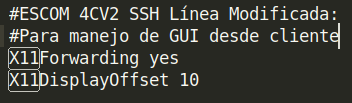
Para finalizar esta configuración debemos habilitar la escritura del archivo en el servicio rsyslog.

Para ello agregamos la siguiente línea al final del archivo /etc/rsyslog.conf



### 2.6 Ejecución de Aplicaciones de GUI de forma Remota.

Para configurar la ejecución de aplicaciones con GUI que estén en el servidor ssh desde el o los clientes, se configuran los siguientes lineamientos:

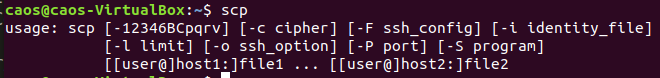


### 2.7 Copiar y Administrar Recursos del Sistema de Archivos de forma Remota.

Para tareas de administración de recursos del sistema de archivos se emplean herramientas del shell del sistema.

Cuando los usuarios tienen acceso al servidor, se les brinda una terminal de comandos donde tienen acceso completo en sus directorios home.

Por ejemplo: Para transferir archivos de un cliente a un servidor, se utiliza la herramienta scp. La sintaxis del comando es la siguiente:



## 3. Brindar servicios locales por SSH.

Consideraremos para esta práctica los servicios remotos.

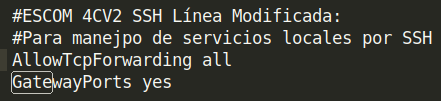
● MYSQL.

Este servicio se ejecuta de manera local en el servidor SSH y escucha peticiones sólo en la interfaz local.

● HTTP.

El servicio http también se ejecuta de manera local en el servidor ssh

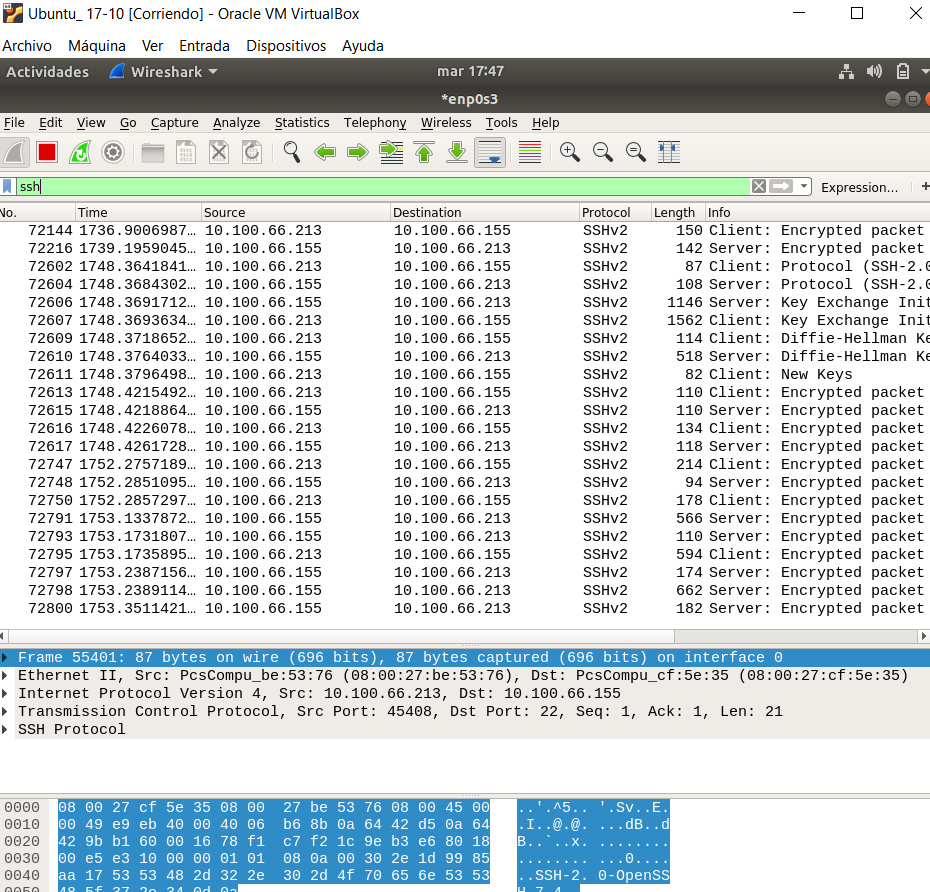
Para logar esta actividad modificaremos en el archivo de configuración:



## 4. Pruebas.

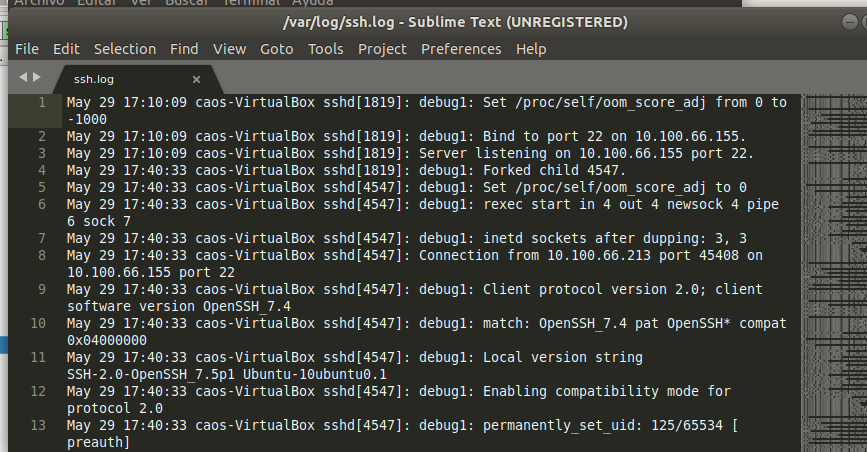
Las pruebas se realizan utilizando el analizador de tramas y los logs del servidor con el nivel de DEBUG, en este caso es mediante WireShark, seleccionando la interfaz y el filtro adecuado y mediante el comando top -i -U nombre de usuario en la terminal.

Las capturas de paquetes del servidor SSH aparecen con el protocolo SSHv2:



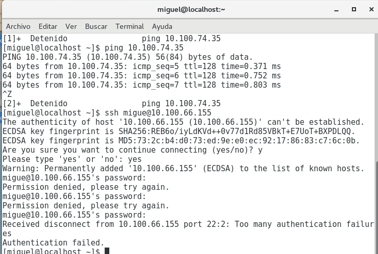
El Log lo podemos revisar tecleando en terminal:

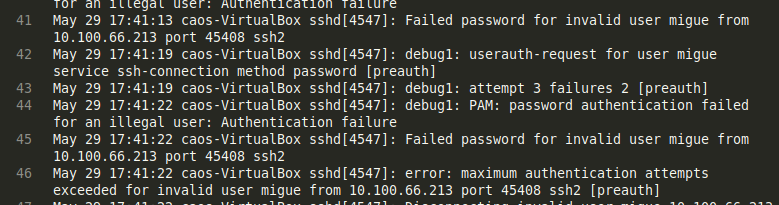
$ subl /var/log/ssh.log



### 4.1 Lista de usuarios y máximo intento de Accesos.

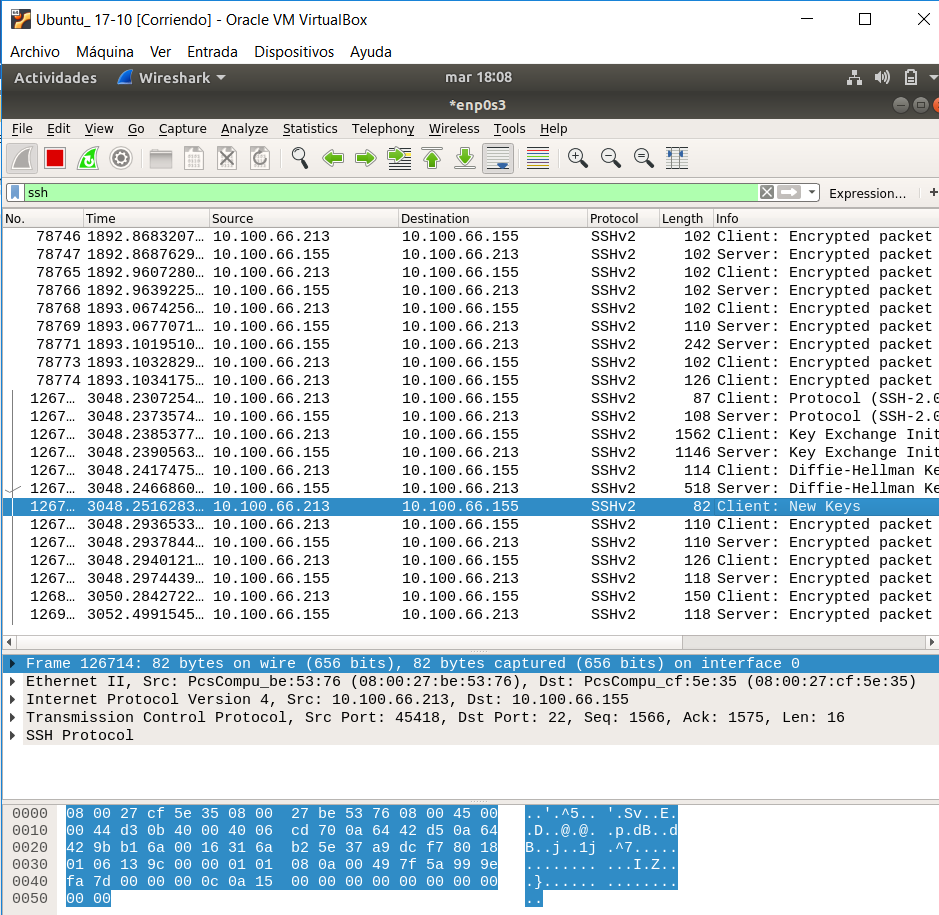
Verificamos que los usuarios que no pertenecen a la lista son rechazados como inválidos y que el máximo número de intentos para login sea 3.

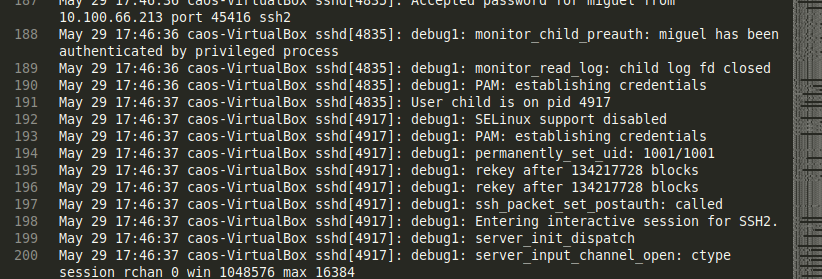




### 4.2 Inicio de sesión por certificado.

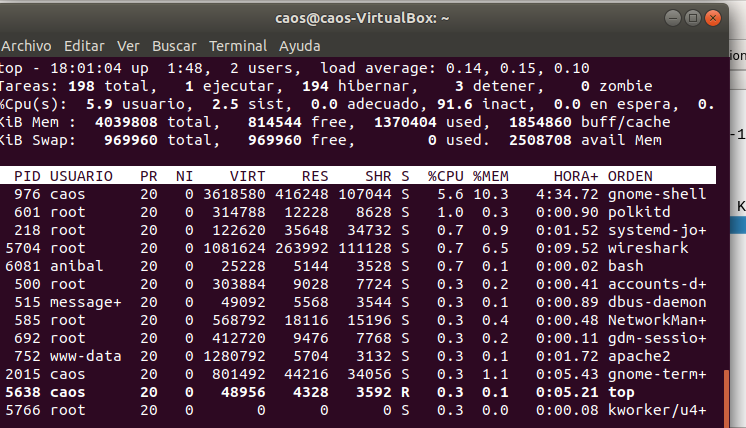
Inicio de sesión: key exchange



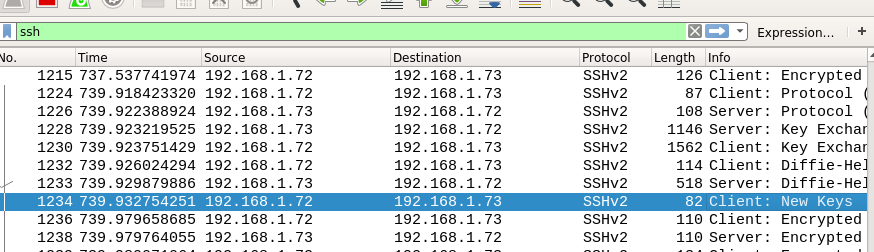


Inicio de sesión por clave RSA (sin contraseña de login)

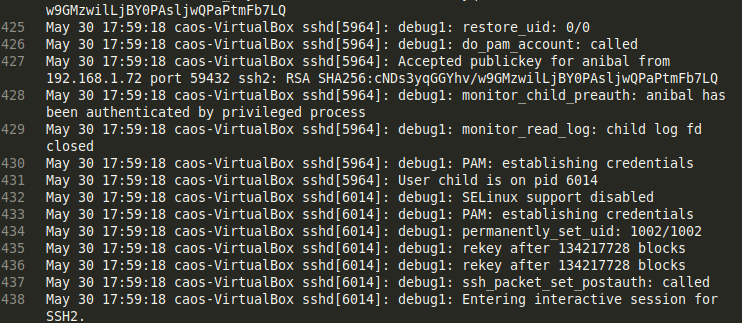
Observamos el Usuario anibal:



Y en el monitor de tramas podemos observar la orden de llave:

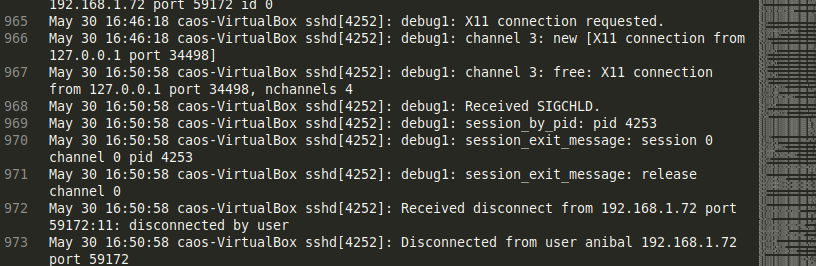


Y en el Log el acceso mediante la llave RSA



### 4.3 Forwarding de aplicaciones gráficas.

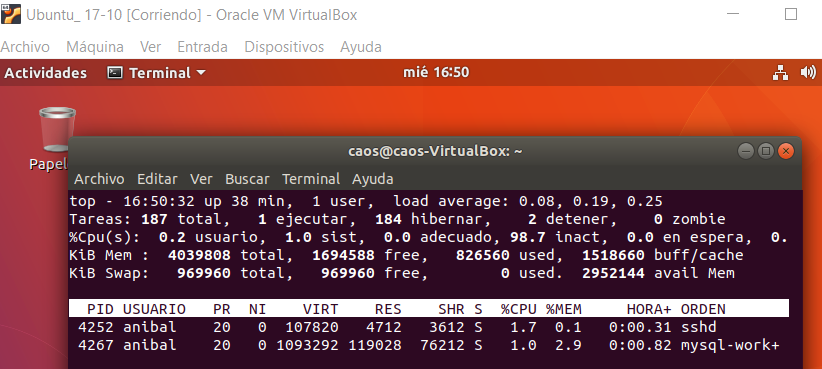
Logs de conexión X11:



Envío de paquetes para cargar aplicación GUI. Capturas de las tramas ssh durante el envío, mediante el comando top en terminal.

$ top -i -U anibal

Y podemos observar cómo el usuario anibal está haciendo uso del programa MYSQL-WORKBENCH, a través de ssh:



Donde en el lado del cliente podemos verificar que:

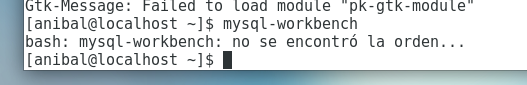
El cliente anibal ejecuta localmente la aplicación MSQL-Workbench la cual no se encuentra instalada en el cliente, solamente en el Servidor, mediante terminal.

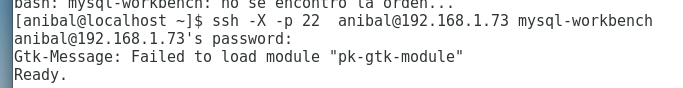
$ ssh -X -p 22 user@servidor aplicación

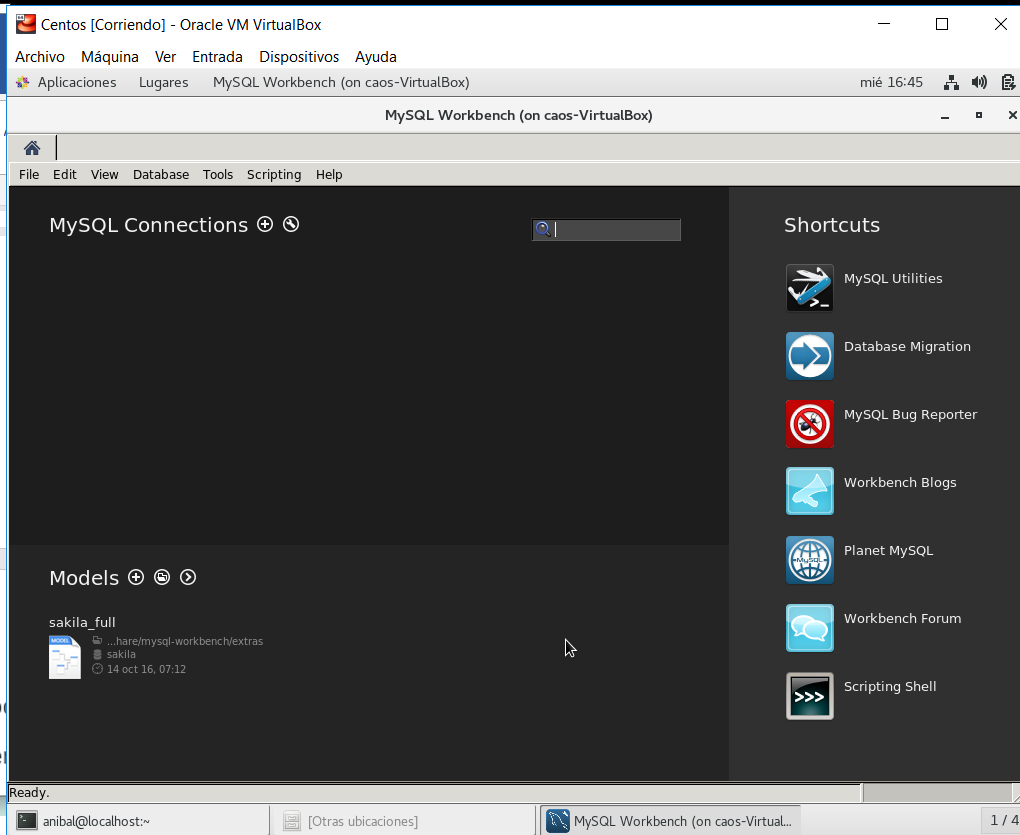
Donde:

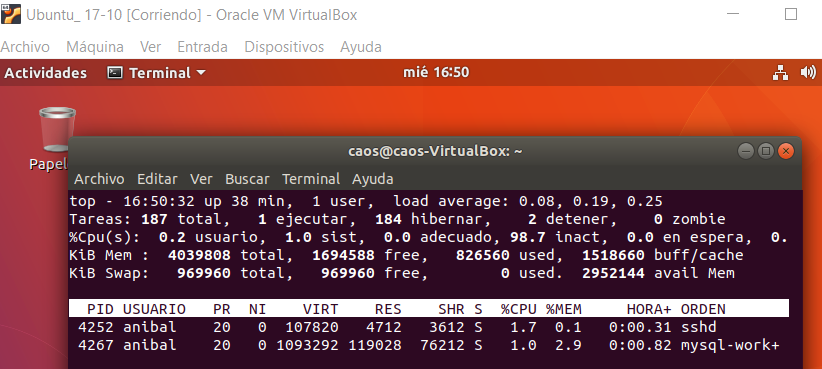
-X = Que vamos a usar interfaz gráfica

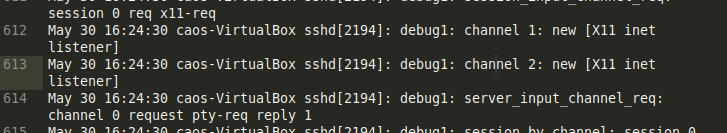
-p = # Puerto





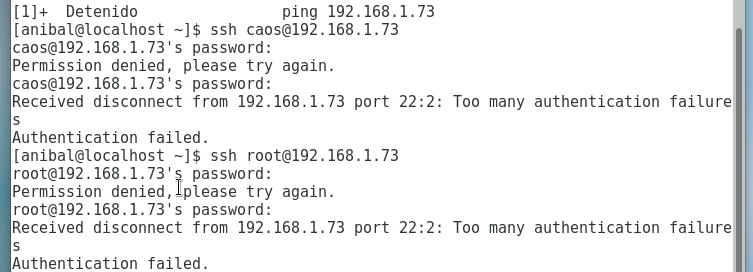


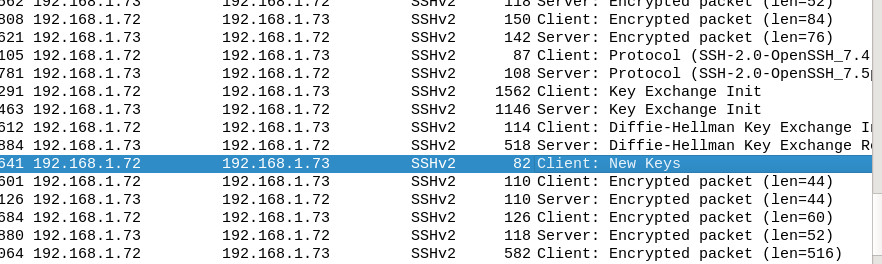




### 4.4 Denegación de usuario root.

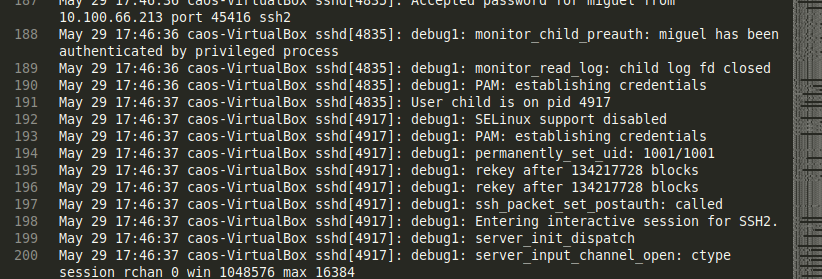
Login prohibido como usuario root:





### 4.5 Inicio de sesión por contraseña.

Inicio de sesión por contraseña:



### 4.6 Servicios por túnel SSH.

Para HTTP:

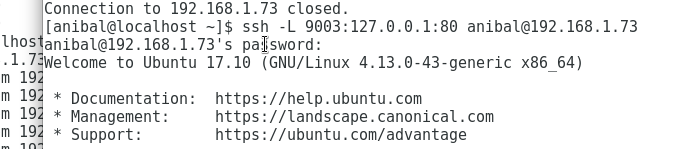
El servidor tiene ahora el puerto 9003 escuchando peticiones HTTP de manera local a través de la petición del cliente mediante terminal:

$ ssh -L 9003:dirección en cliente:80 usuario@servidor

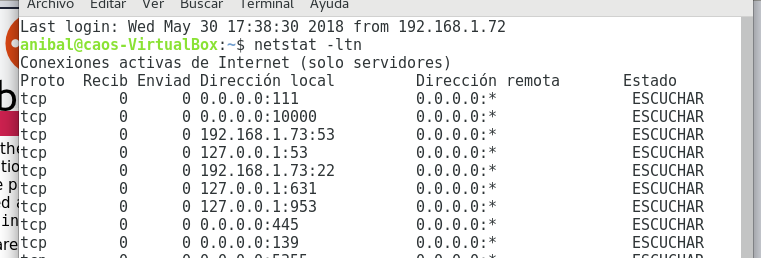
Donde:

9003 = Puerto Local

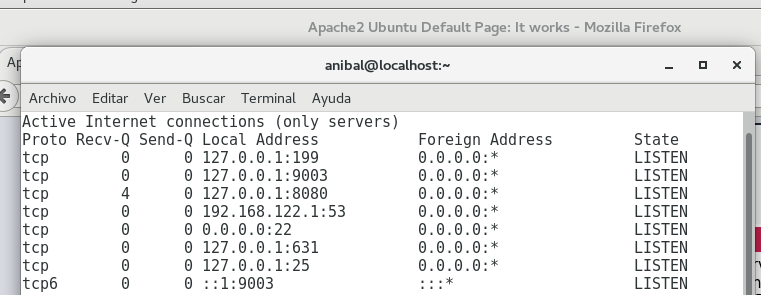
80= Puerto en Servidor



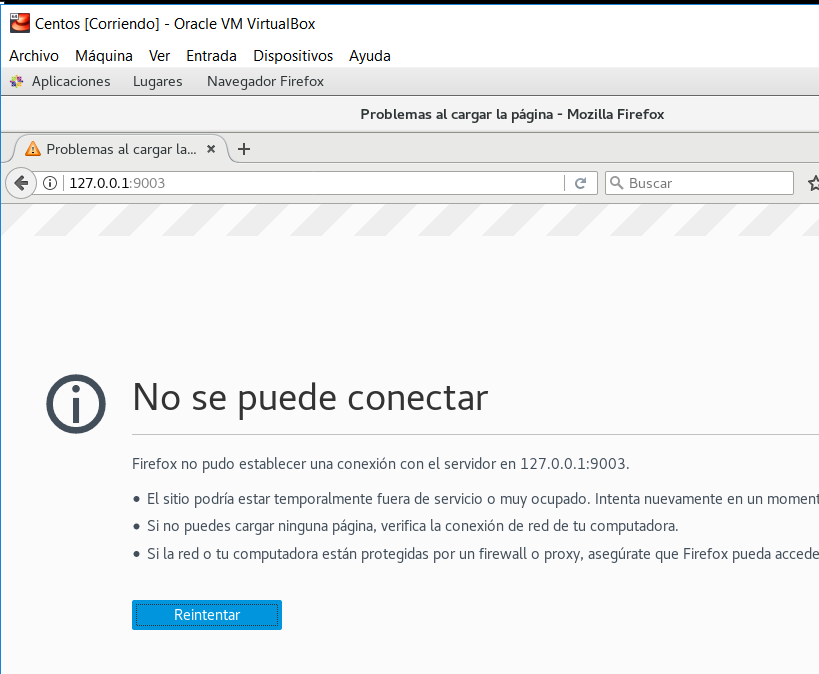
Podemos verificar la escucha en puerto 22:

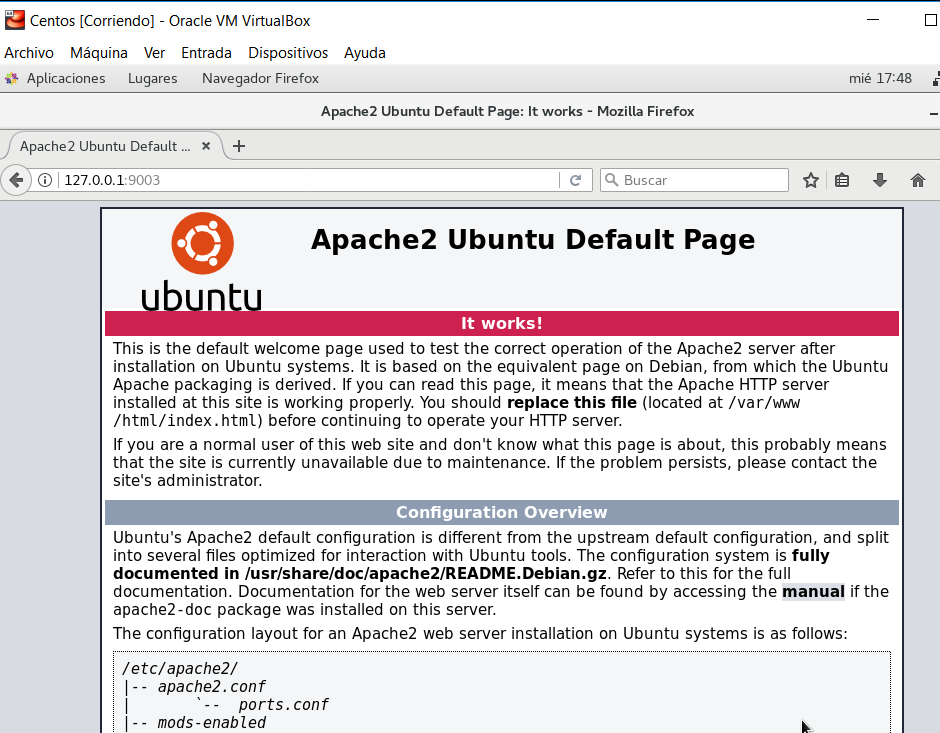


Y el enlace en el cliente a través del puerto 9003:



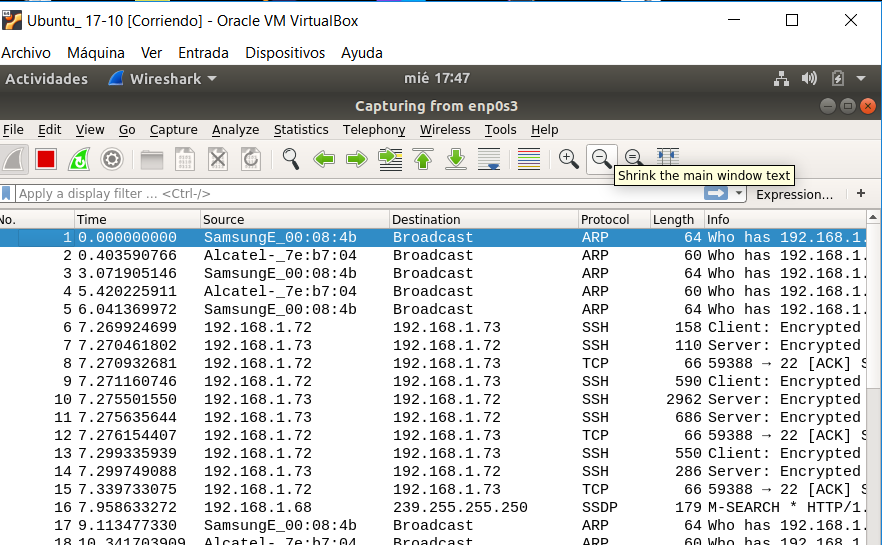
En la máquina del cliente accede al servicio HTTP del servidor vía SSH vía local por el puerto 9003.



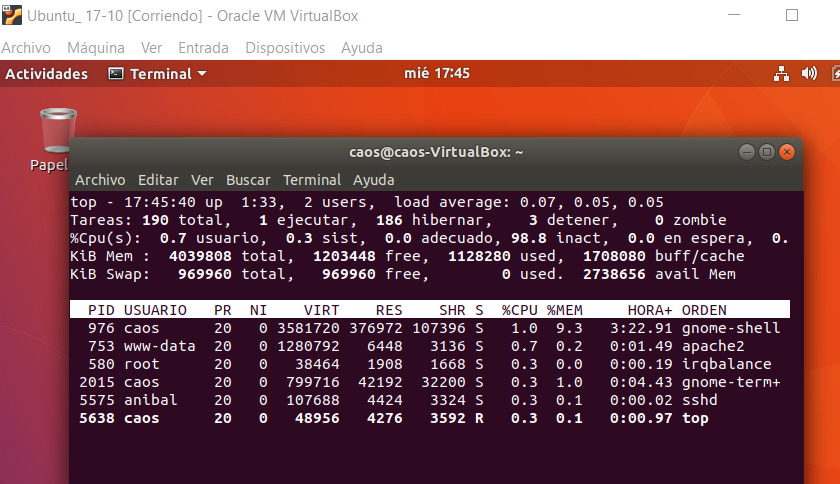


Y lo podemos verificar en la comunicación entre las direcciones IP:

192.168.1.73 (Servidor) y 192.168.1.72 (cliente).



Además podemos verificar la orden que llama al servidor apache, vía el usuario anibal en ssh:



5. Referencias.

[1] Configuración general. Disponible en:

<https://www.openssh.com/manual.html>

[2] Llaves. Disponible en:

<http://man.openbsd.org/ssh-keygen.1>