**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**

**Cliente OpenSSH.**

[**1. Instalación.**](#_gjdgxs) **4**

[UBUNTU:](#_30j0zll) 4

[CENTOS:](#_1fob9te) 5

[**2. Inicio de Sesión.**](#_3znysh7) **10**

[2.1. Por cuenta](#_2et92p0) 10

[2.2. Por llave](#_tyjcwt) 11

[**3. Acciones Básicas.**](#_3dy6vkm) **14**

[3.1. Conexión al servidor mediante interfaz gráfica](#_1t3h5sf) 14

[3.2. Copia y administración de recursos del sistema de archivos de forma remota.](#_4d34og8) 16

[3.3. Conexión a los servidores de aplicación.](#_2s8eyo1) 20

[**4. Referencias**](#_17dp8vu) **23**

## 1. Instalación.

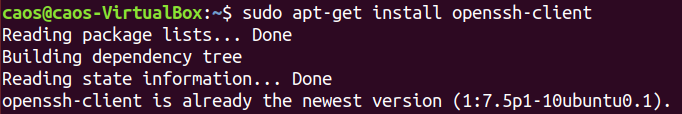
Trabajaremos en la versión 17.10 de Ubuntu (SERVIDOR) y la versión 7 de CentOS a través del Hypervisor VirtualBox desde una laptop Lenovo Y700 con Sistema Operativo Windows 10 Home Edition.

Para instalar los paquetes de openssh-client abriremos una ventana de terminal (CTRL + ALT +T).

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

### UBUNTU:

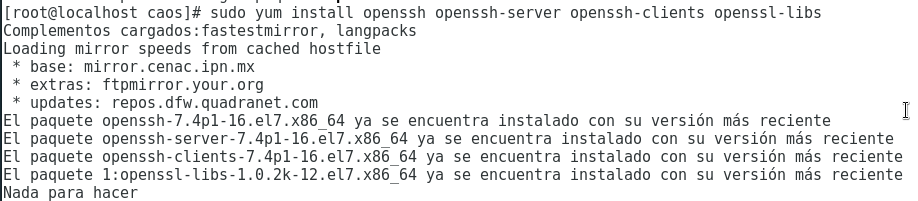
$ sudo apt-get install openssh-client



### CENTOS:

Como root:

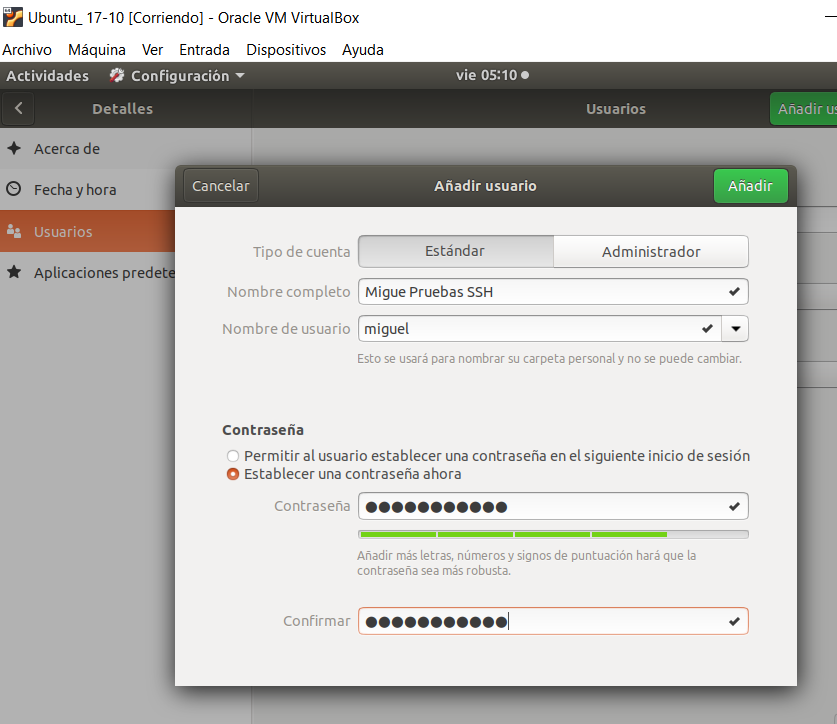
# yum install openssh openssh-server openssh-clients openssl-libs



Se crea en UBUNTU el usuario Migue Pruebas SSH con los siguientes parámetros:

Usuario: miguel

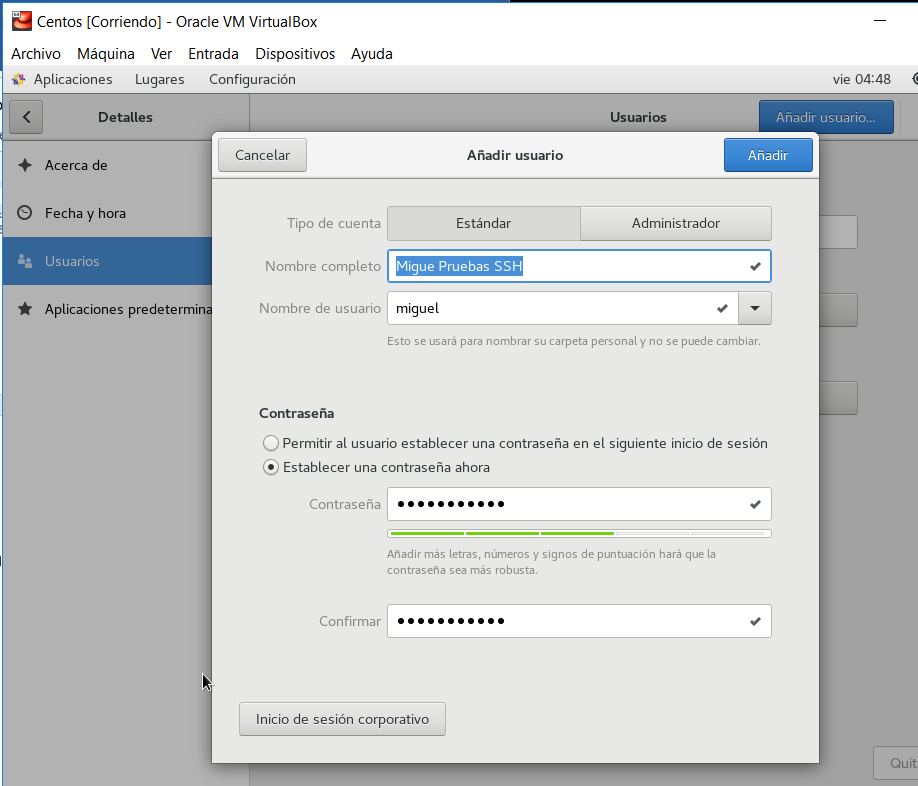
Password: MigueSsh123



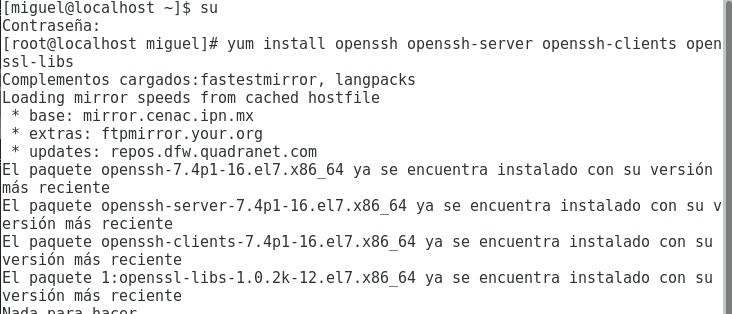
Se crea en CentOS el usuario Migue Pruebas SSH con los siguientes parámetros:

Usuario: miguel

Password: MigueSsh123



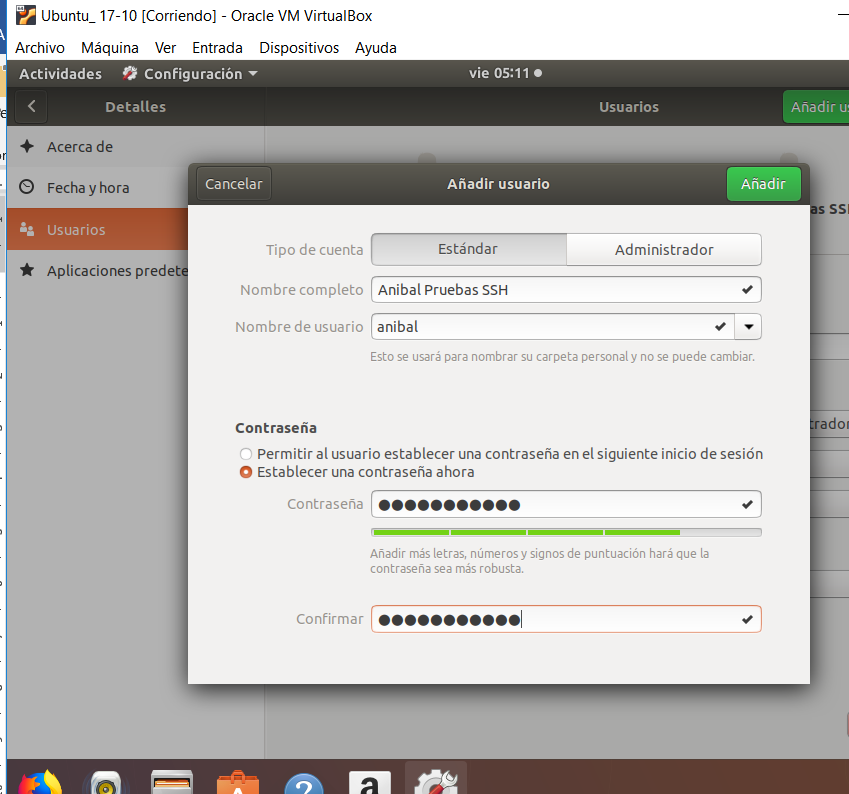
Verificamos que todo esté debidamente instalado:



Se crea en UBUNTU el usuario Aníbal Pruebas SSH con los siguientes parámetros:

Usuario: anibal

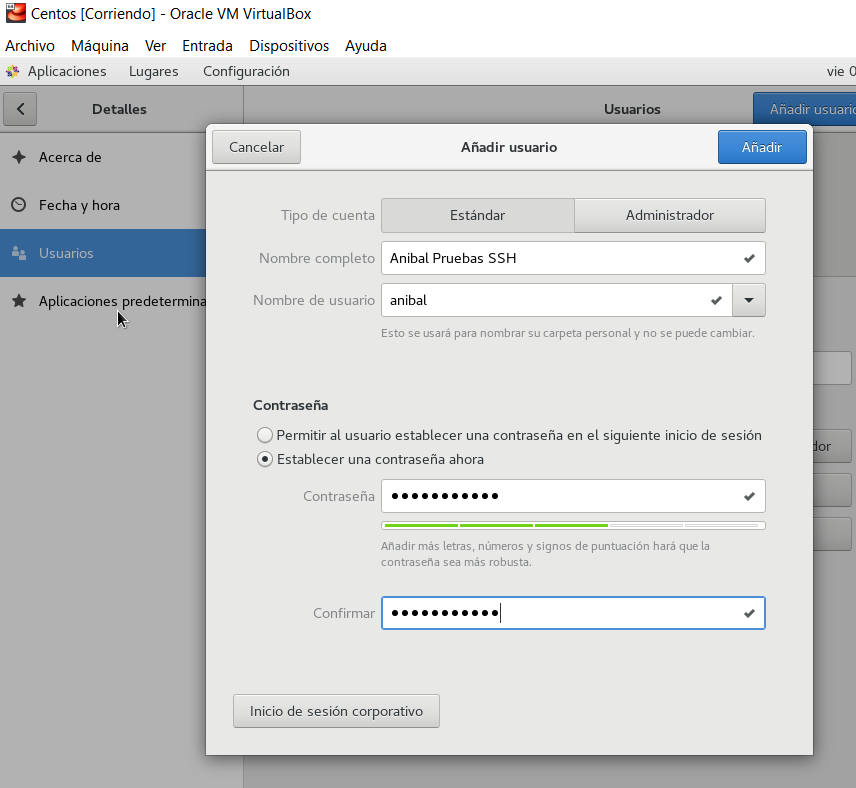
Password: AnibaSsh123



Se crea en CENTOS el usuario Aníbal Pruebas SSH con los siguientes parámetros:

Usuario: anibal

Password: AnibaSsh123



## 2. Inicio de Sesión.

### 2.1. Por cuenta

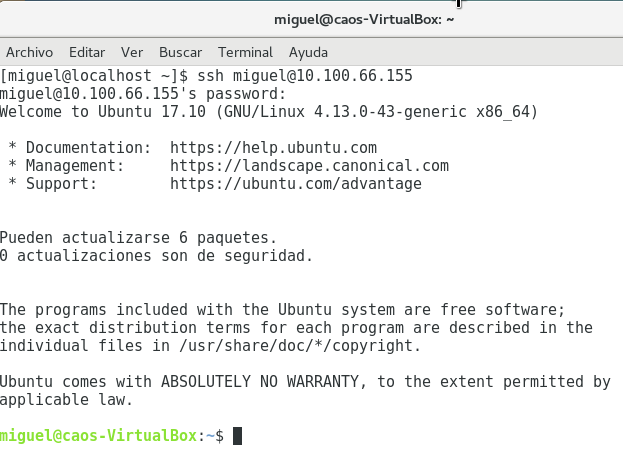
Para acceder mediante una cuenta, se requiere que esta cuenta ya esté creada y por lo tanto el administrador del servidor SSH es la entidad que brinda los datos: usuario y contraseña.

Procedimiento.

1. Abrir una terminal y escribir: ssh usuario@direcciónIP , tal cual como se muestra en la imagen:

2. Ingresar Contraseña.

3. Accesará al SO Ubuntu 17.10 desde la terminal del SO CentOS 7 del Cliente:



### 2.2. Por llave

Para acceder mediante una llave, se requiere que el usuario cree los archivos correspondientes a esta llave y el administrador del servidor cuente con este archivo creado.

Creación de llave

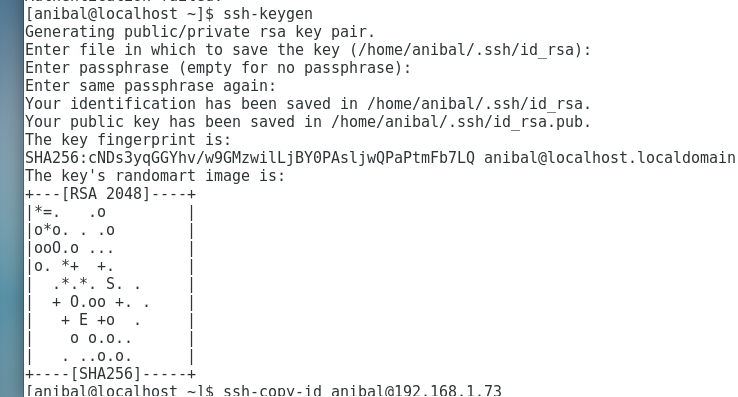
1. Abrir una terminal e introducir la siguiente sentencia:

$ ssh-keygen

2. Observamos que nos pide la ruta donde se crearan las llaves.

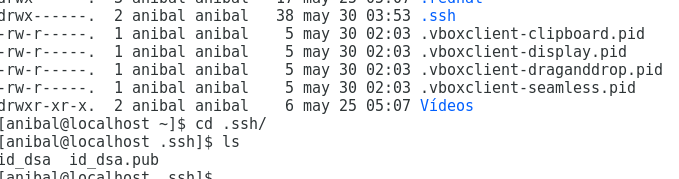
Damos enter para indicar que haga la creación de llaves en la ruta predeterminada.

Esperamos a la creación de la llave:



La clave de la llave es: anibal (la cual es diferente de la clave oiriginal: AnibaSsh123)

Y podemos verificar su creación:



Para copiar esta llave al servidor desde terminal tecleamos:

$ ssh-copy-id anibal@192.168.1.73

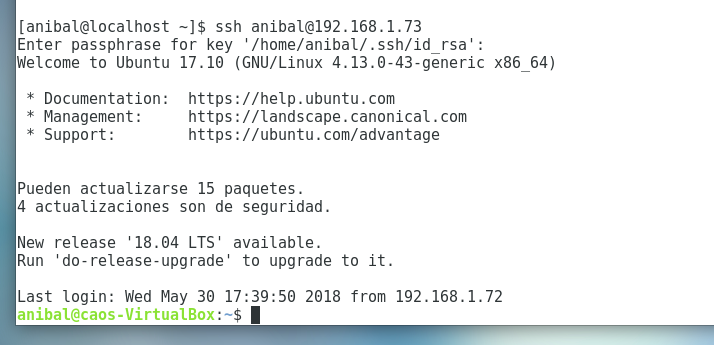
Donde:

anibal = usuario

192.168.1.73 = Dirección de Servidor

Probamos la conexión desde terminal tecleamos:

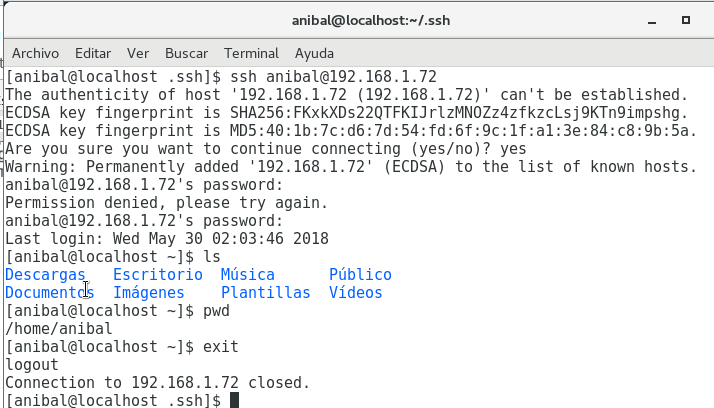
$ ssh anibal@192.168.1.73



Inicio de sesión.

Al tener la llave y que dicho archivo lo contenga el administrador del servidor, él nos debe proporcionar nuestra cuenta de usuario (en este caso no contamos con ninguna contraseña, ya que para nosotros nuestra autenticación es la llave), realizamos los siguientes pasos:

1. Abrimos una terminal y escribimos lo siguiente: ssh usuario@direcciónIP, así como se muestra en la imagen:



## 3. Acciones Básicas.

### 3.1. Conexión al servidor mediante interfaz gráfica

Podemos ejecutar una aplicación que no tenemos en el equipo actual teniendo esta aplicación instalada en el equipo remoto y trabajar localmente. A continuación, se describen los pasos:

Ejemplo si deseamos abrir MYSQL-WORKBENCH la instrucción quedaría de la siguiente forma, como se muestra en la imagen:

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

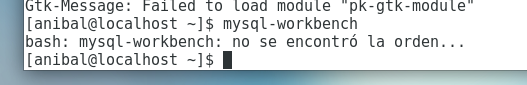
El cliente anibal ejecuta localmente la aplicación MSQL-Workbench la cual no se encuentra instalada en el cliente, solamente en el Servidor, mediante terminal.

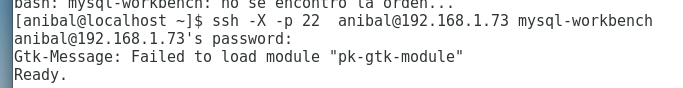
$ ssh -X -p 22 user@servidor aplicación

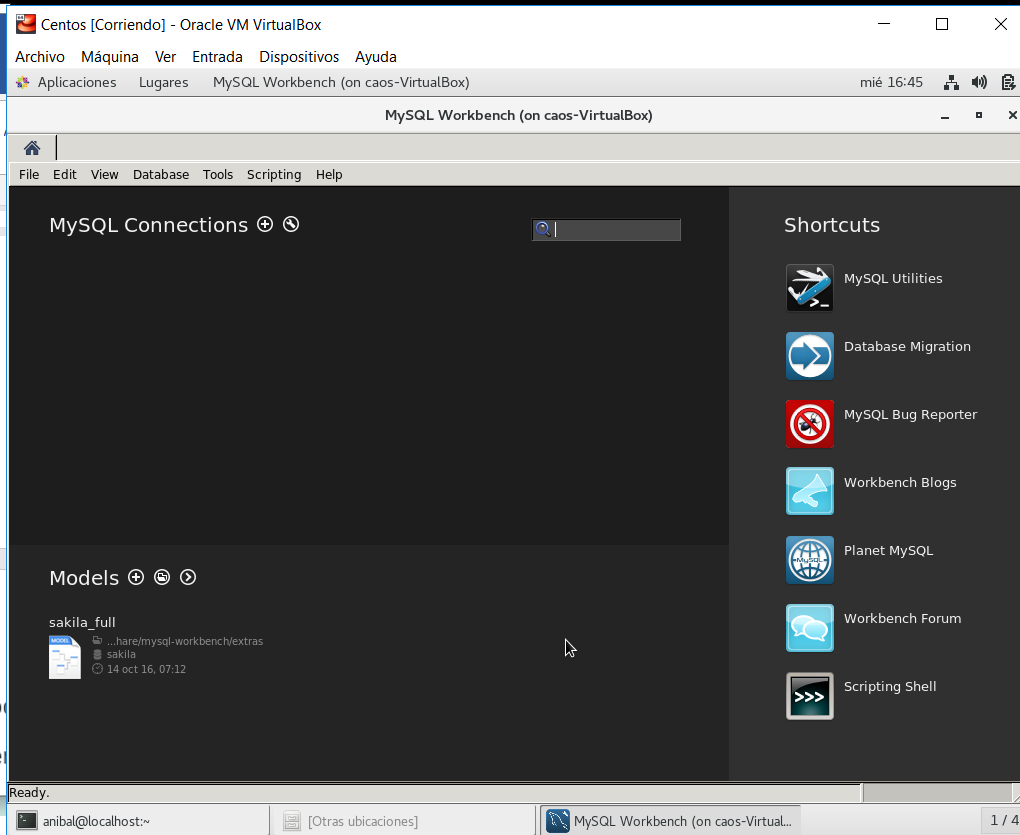
Donde:

-X = Que vamos a usar interfaz gráfica

-p = # Puerto



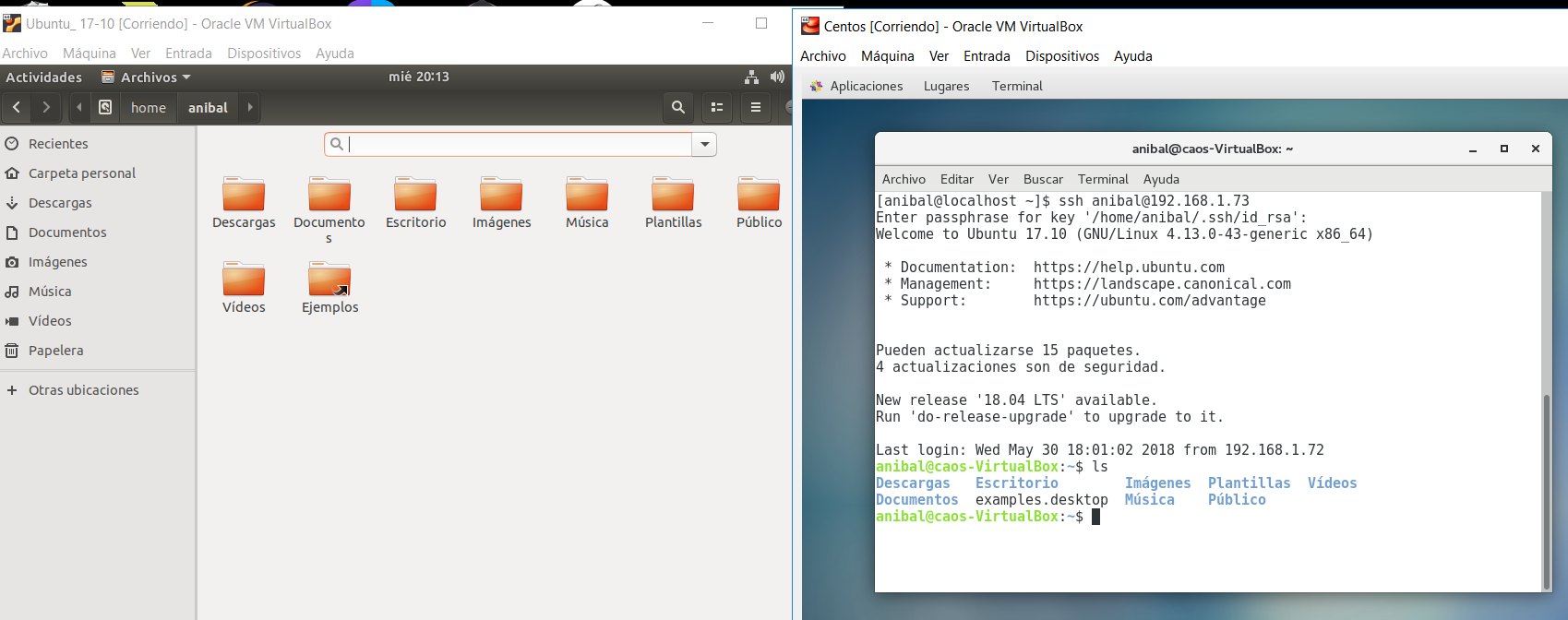




### 3.2. Copia y administración de recursos del sistema de archivos de forma remota.

Para realizar transferencia de archivos, hacemos uso del comando scp. El comando scp puede ser usado para transferir archivos entre máquinas sobre una conexión encriptada y segura.

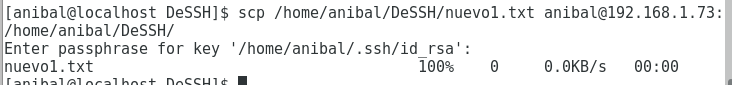
Antes de envío de archivos:



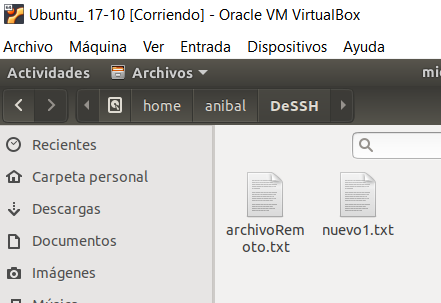
Transferir archivo a nuestro sitio remoto ubicado en el servidor

1. Abrimos una terminal y escribimos la siguiente línea:

$ scp rutaArchivo usuario@ipServidor:/nombreNuevoArchivo



Podemos verificar en Carpeta del Servidor que nuestro archivo fue copiado con éxito.

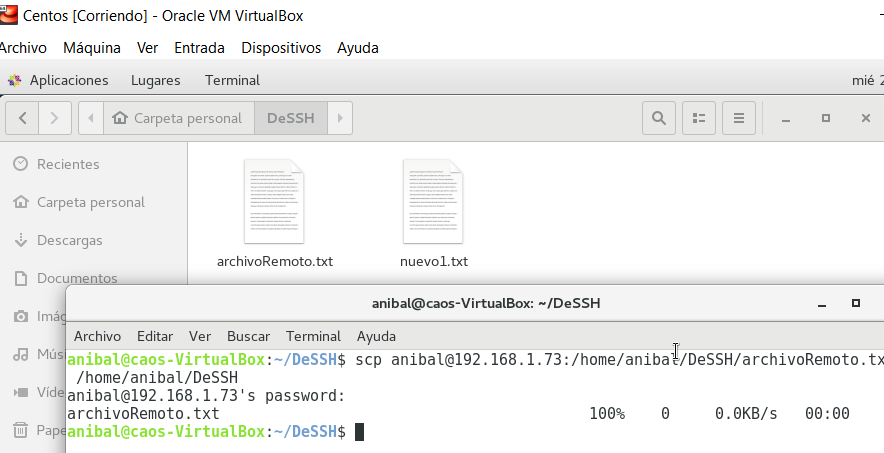


Transferir archivo ubicado remotamente en el servidor a nuestra máquina

1. Abrimos una terminal y escribimos la siguiente línea:

$ scp usuario@ipServidor:/rutaArchivoRemoto rutaLocal

Ejemplo:



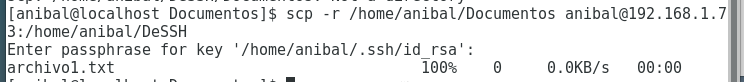
Transferencia de múltiples archivos

Se pueden especificar múltiples archivos si tomamos como base un directorio. Por ejemplo, para transferir el contenido de un directorio a otro en la máquina remota (donde se encuentra el servidor ssh) realizamos lo siguiente:

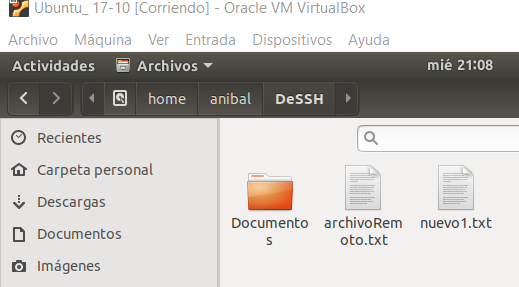
1. Abrimos una terminal y escribimos la siguiente línea:

$ scp -r rutaLocalDirectorio/ usuario@ipServidor:/rutaDirectorioRemoto

Ejemplo:



Verificamos en el Servidor:



### 3.3. Conexión a los servidores de aplicación.

Conexión al servidor de aplicación HTTP.

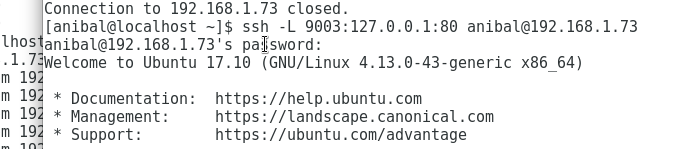
El cliente se comunicará con el servidor a través del puerto 9003 escuchando peticiones HTTP de manera local mediante terminal:

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

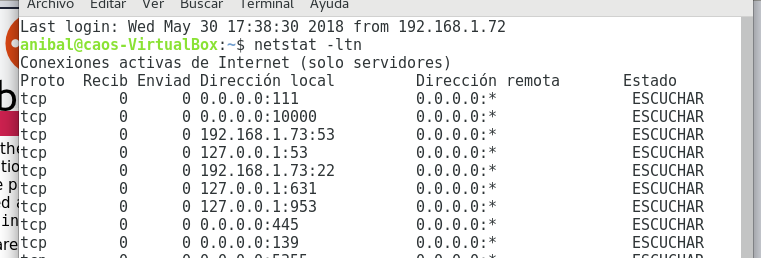
Donde:

9003 = Puerto Local

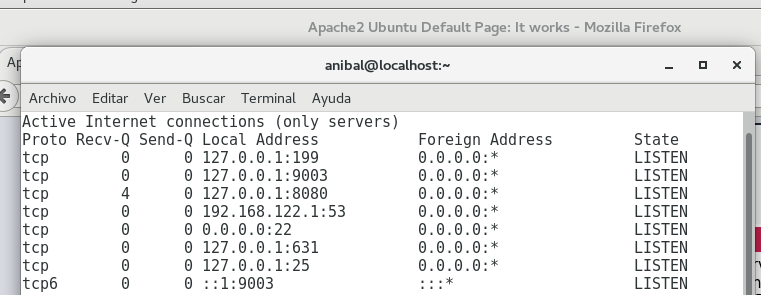
80= Puerto en Servidor



Podemos verificar la escucha en puerto 22:

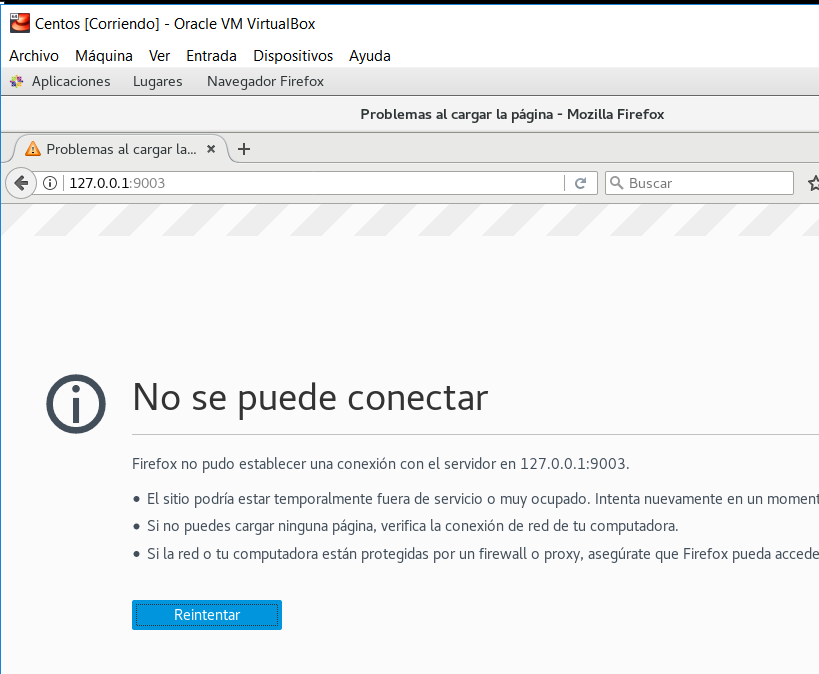


Y el enlace en el cliente a través del puerto 9003:

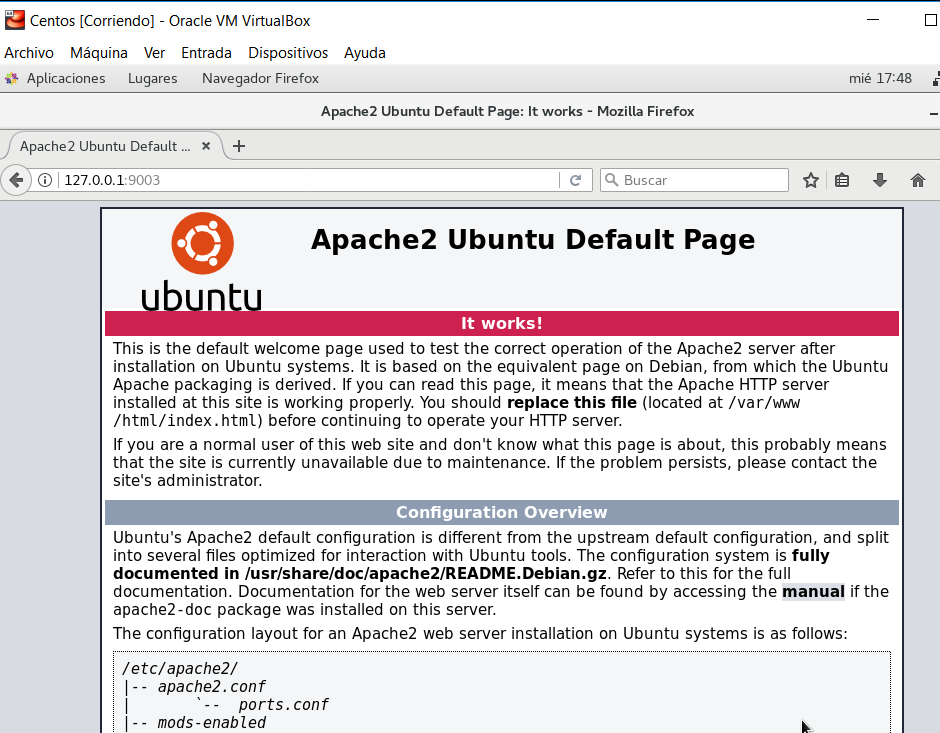


En la máquina del cliente accede al servicio HTTP del servidor vía SSH vía local por el puerto 9003.

Podemos observar la dirección Web antes de la comunicación SSH:



Y posterior a la comunicación SSH:



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

192.168.1.73 (Servidor) y 192.168.1.72 (cliente).

## 4. Referencias

[1] Configuración general. Disponible en:

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

[2] Uso de scp con ssh. Disponible en:

<http://web.mit.edu/rhel-doc/3/rhel-sag-es-3/s1-openssh-client-config.html>

[3] Creación de llaves. Disponible en:

<https://www.linuxtotal.com.mx/index.php?cont=info_seyre_010>