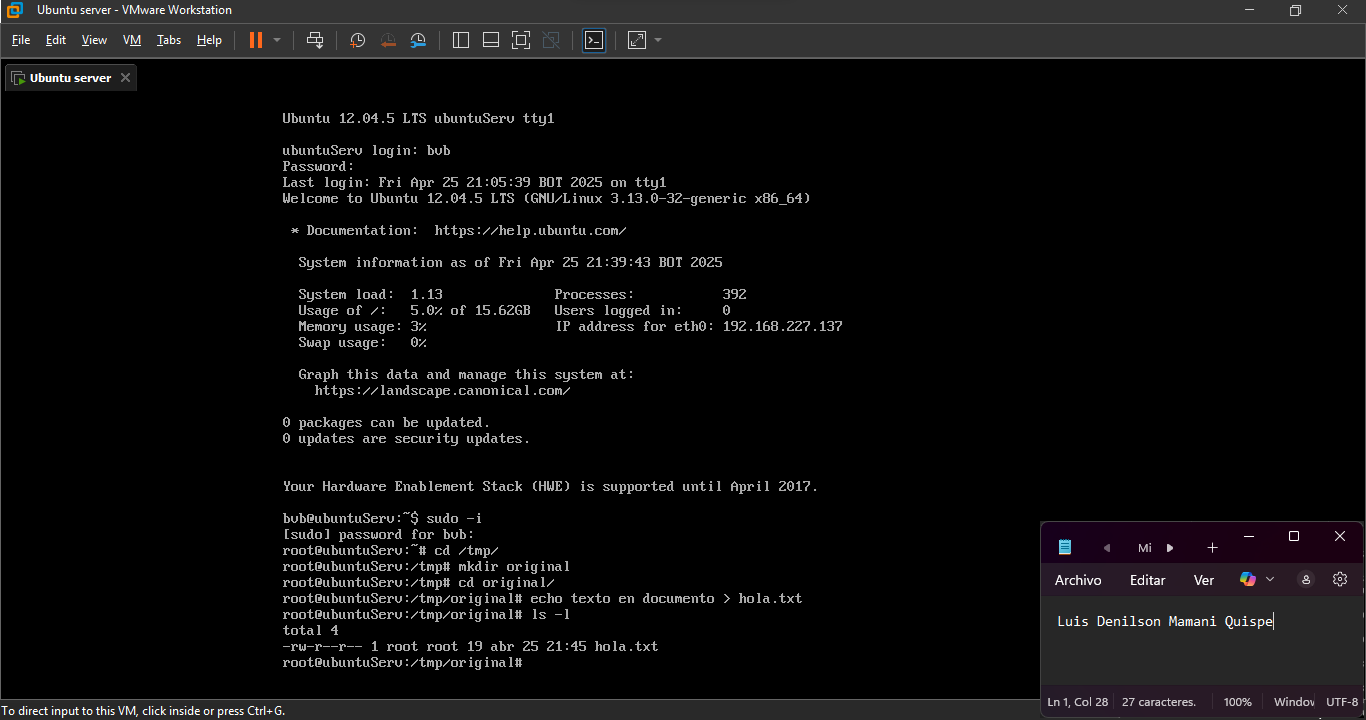
|  |  |  |
| --- | --- | --- |
| LABORATORIO N ° 7  SEGURIDAD DE SISTEMAS SIS-737S1 | |  |
| Estudiante: Luis Denilson Mamani Quispe RU: 103561  Fecha de Entrega: 25/04/2025 | |
| Docente: Ing. J. Alexander Duran M. | Auxiliar: Univ. Aldrin Perez Miranda |

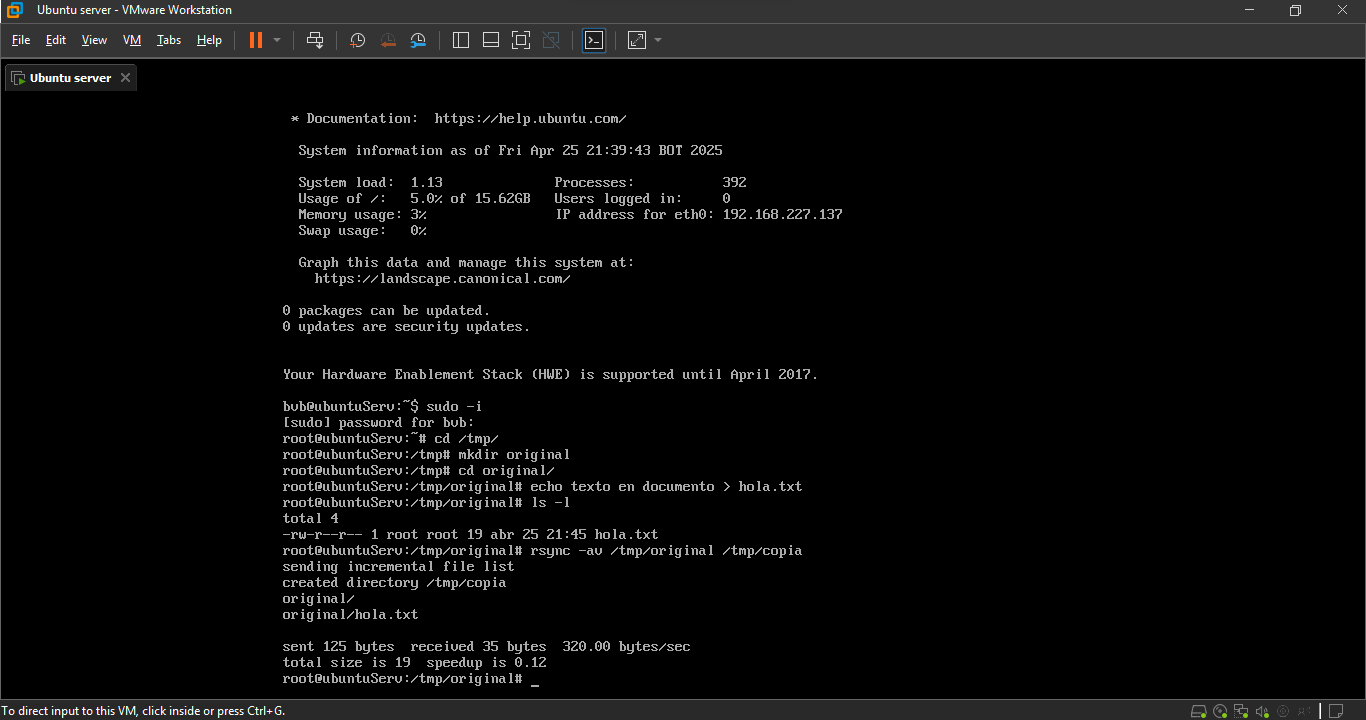
**Desarrollo**

Entramos en nuestro Ubuntu Server 12.04 y nos ponemos en modo super-usuario con **sudo-i**

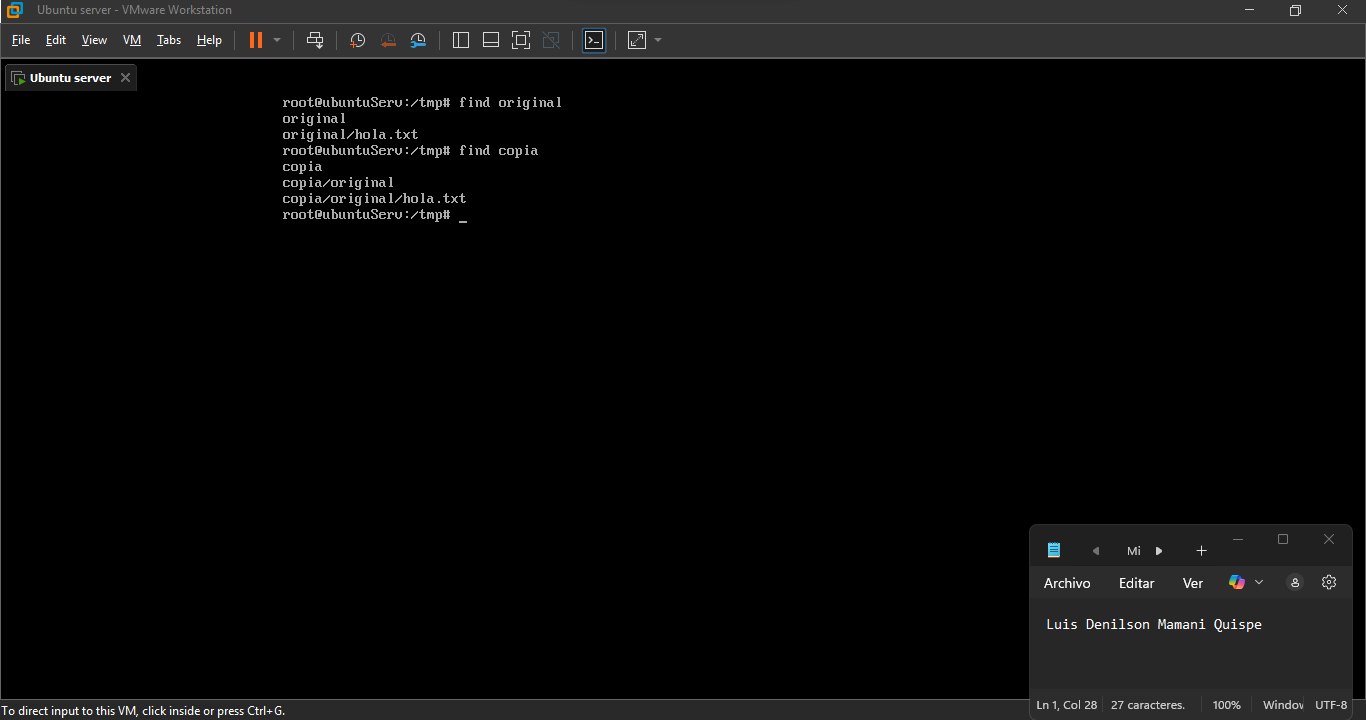
Nos vamos al **/tmp** y creamos un directorio llamado original con un fichero **hola.txt**



Ahora creamos una copia de original mediante el comando: **#rsyn -av /tmp/original /tmp/copia**



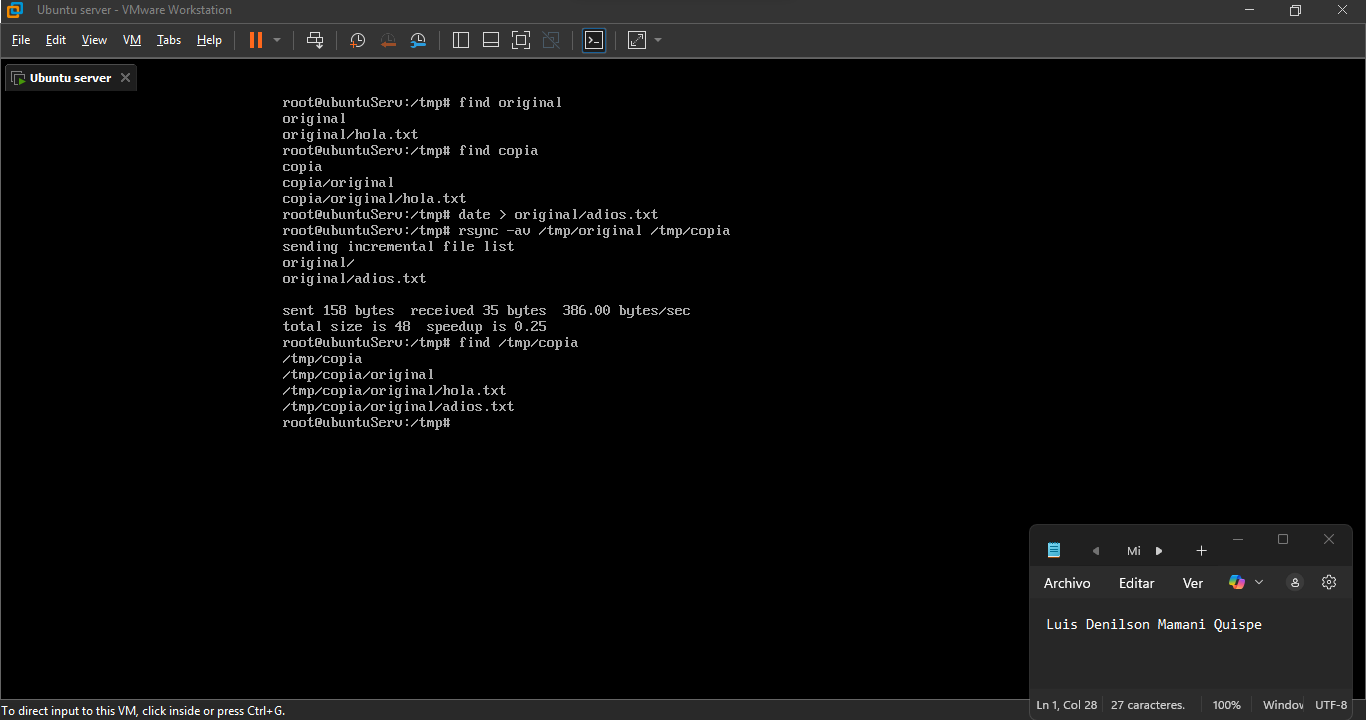
Como se ve en la figura, la herramienta nos avisa de que va a crear el directorio /tmp/copia (no lo habíamos creado) y muestra los ficheros que ha traspasado y un resumen de bytes transferidos.



﻿

El directorio copia reproduce la estructura de carpetas de original, no solo los ficheros.

¿En qué se diferencia de hacer una copia normal mediante cp? Pues en que rsync no copia todo, sino solo los ficheros nuevos o los que han cambiado. Por ejemplo, creamos un fichero nuevo llamado "adios" y sincronizamos. Solo se traspasa ese fichero Usamos **find** para visualizar el contenido



﻿

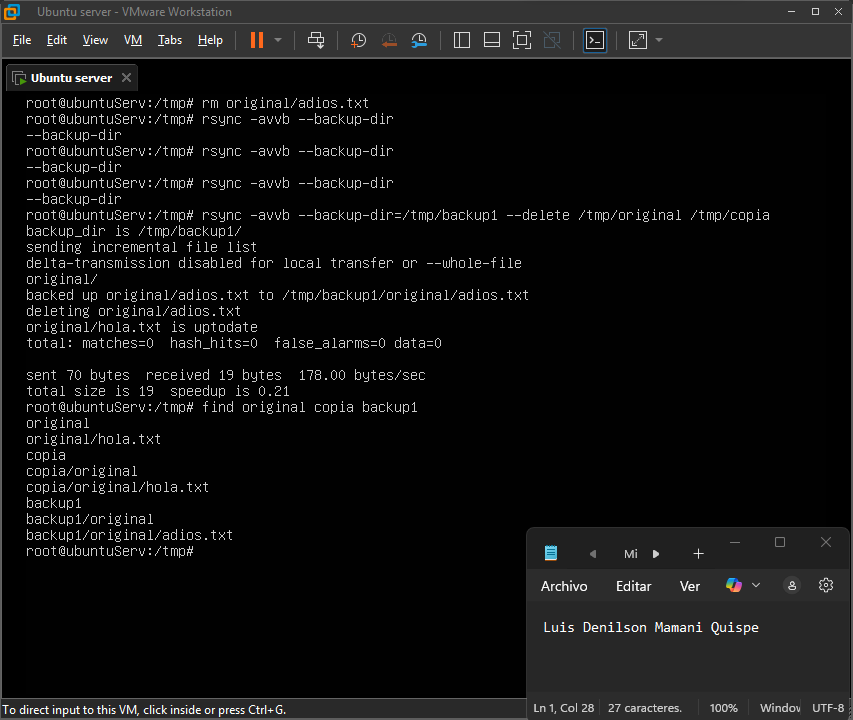
Si hemos borrado un fichero en el original y queremos que se actualice la copia, hay que incluir el parámetro --delete

**PARTE 2**

Con lo que hemos visto hasta ahora solo podemos hacer backups completos. El directorio copia lo podemos llevar a cualquier dispositivo extraíble o podría ser un disco en red. Para hacer backups incrementales ejecutaremos los siguientes comandos:

**# rsync --avvb --delete --backup dir=/tmp/backup1 /tmp/original/tmp/copia**

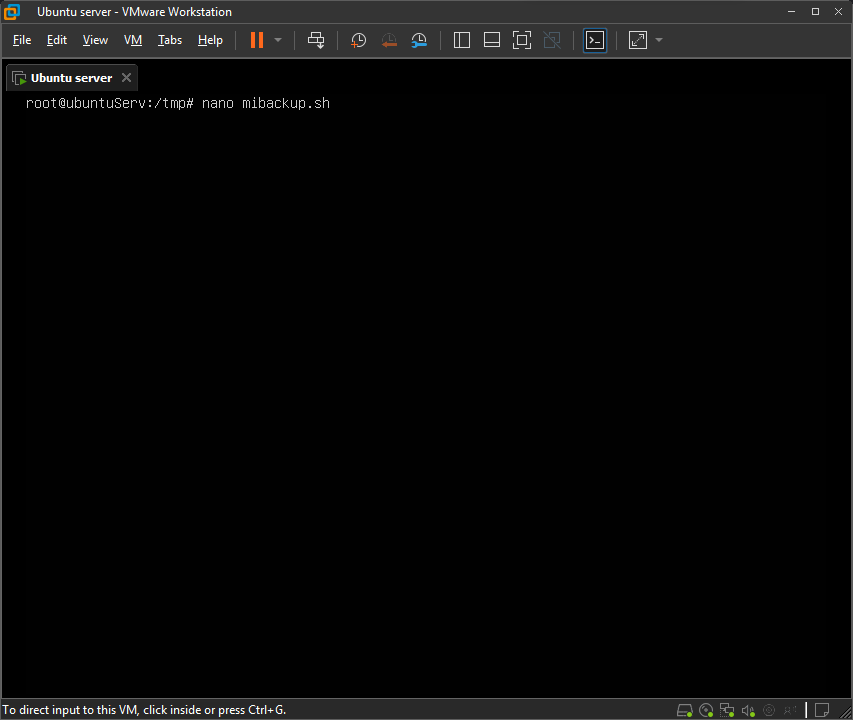
Esta vez la sincronización deja en el directorio **/tmp/backup1** los ficheros que resultan modificados o eliminados; en **/tmp/copia** siempre está la versión actual. En nuestro ejemplo vamos a borrar el fichero adios.txt y al sincronizar vemos que ya no está en original ni en copia, pero sí en backup1

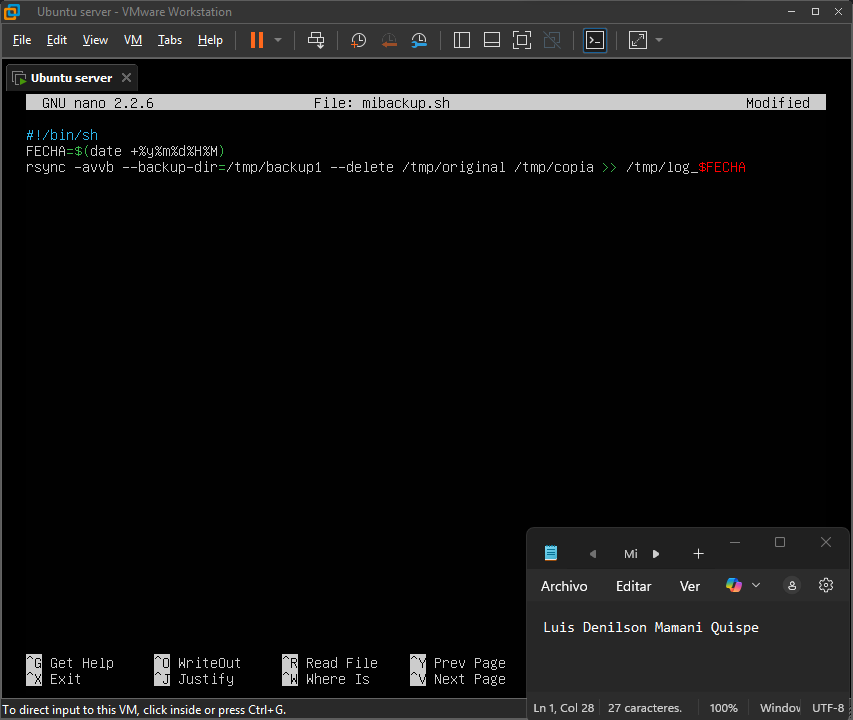


Finalmente, como es imprescindible que el backup se ejecute con regularidad, vamos a probar a meterlo en cron. Le pondremos que se ejecute cada minuto (lo normal sería una vez al día), y en ese tiempo haremos cambios para comprobar el funcionamiento.

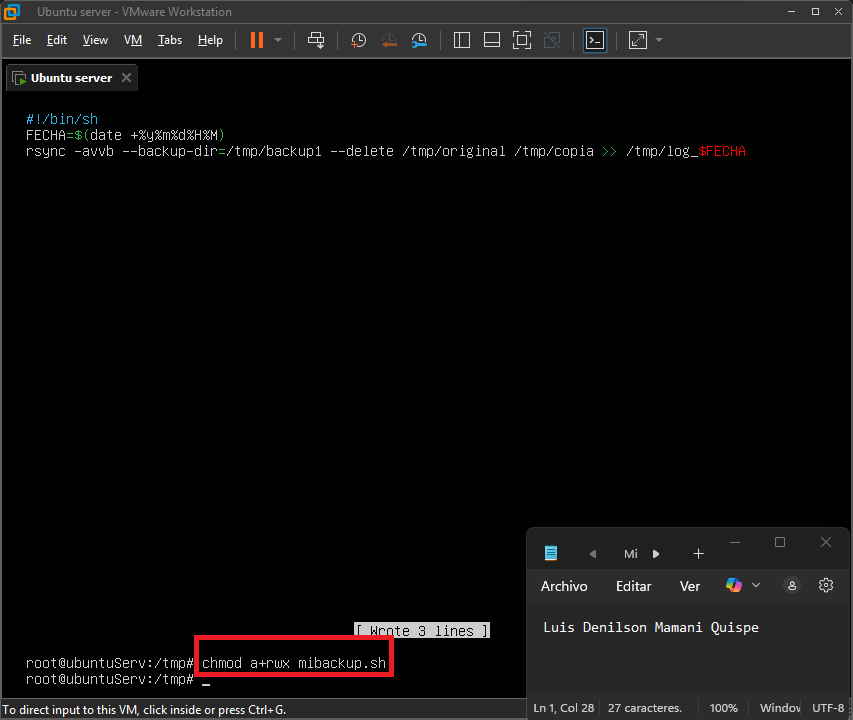
Crearemos un script llamado mibackup.sh (darle permisos de lectura escritura y ejecución) que invocaremos desde el cron. Para distinguir las distintas copias incrementales, el script utiliza la fecha en que se ejecuta, además que dejaremos un log para comprobar las copias.

El script deberá contener el siguiente código (no olvide guardar):

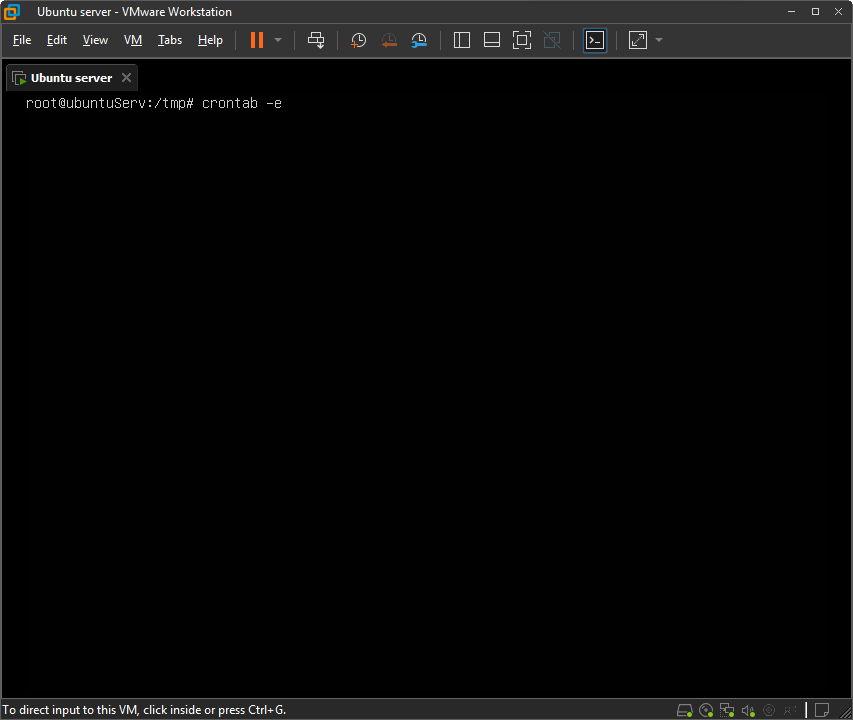


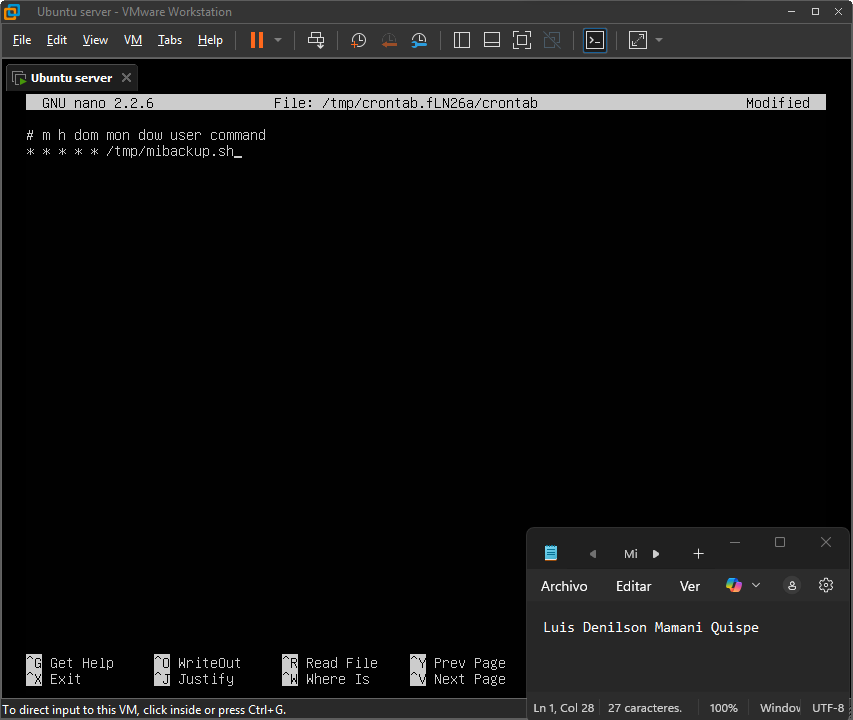


Permisos

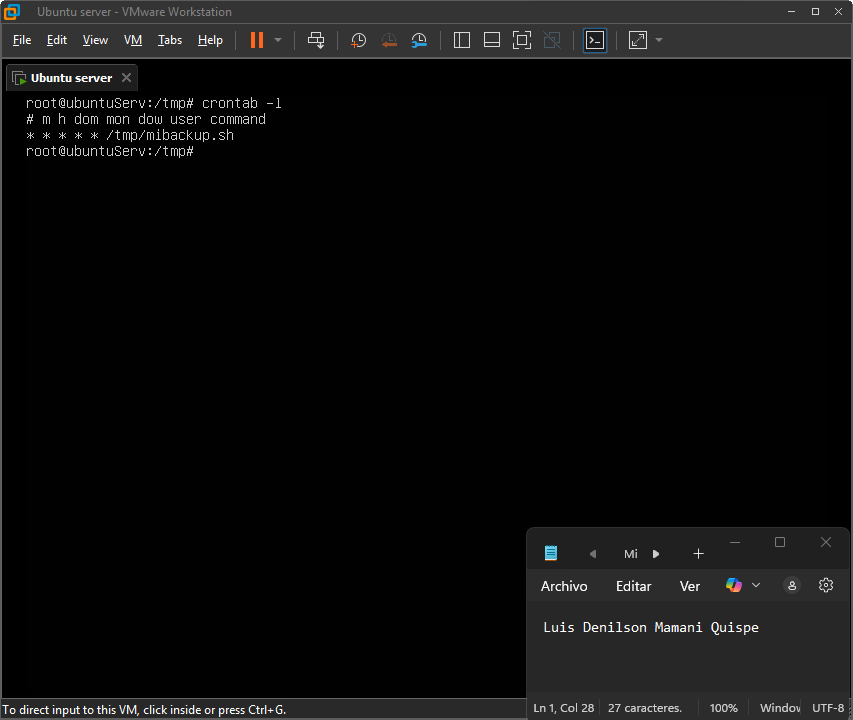


Procedemos a editar el cron ingresando el comando: **crontab -e**

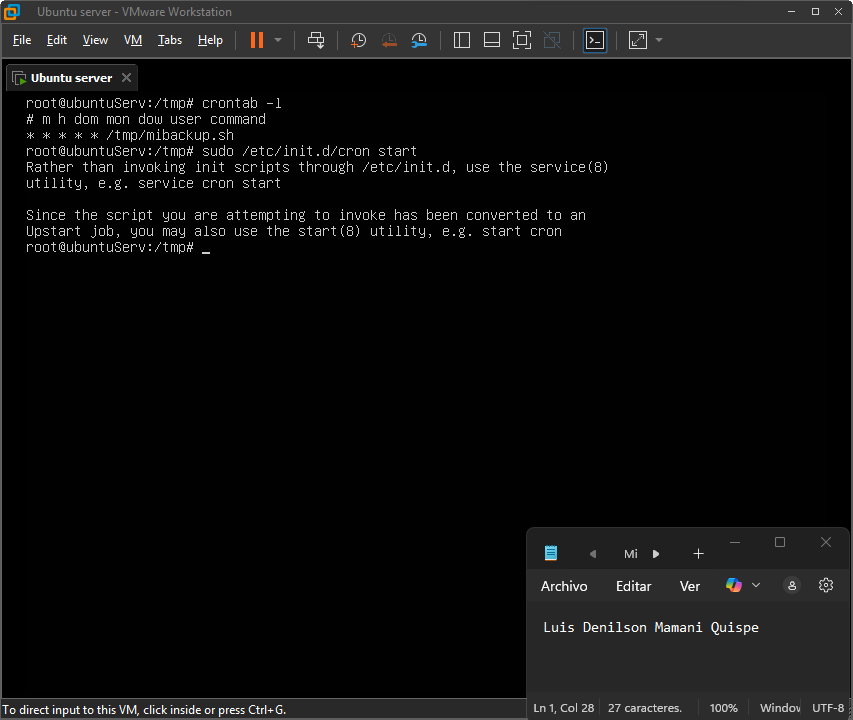




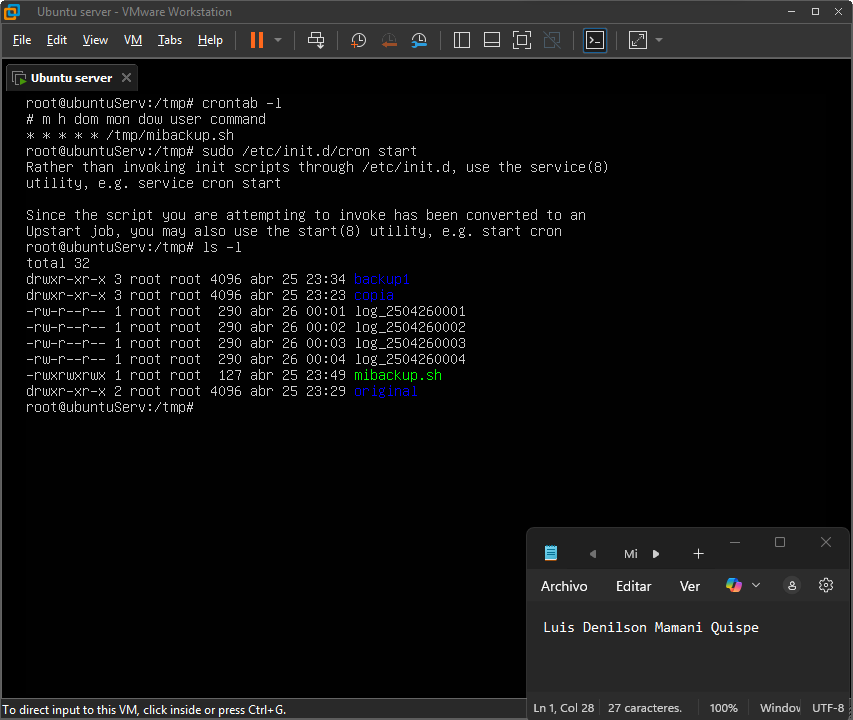
Si queremos ver el contenido de crontab utilizamos: **crontab -l**

****

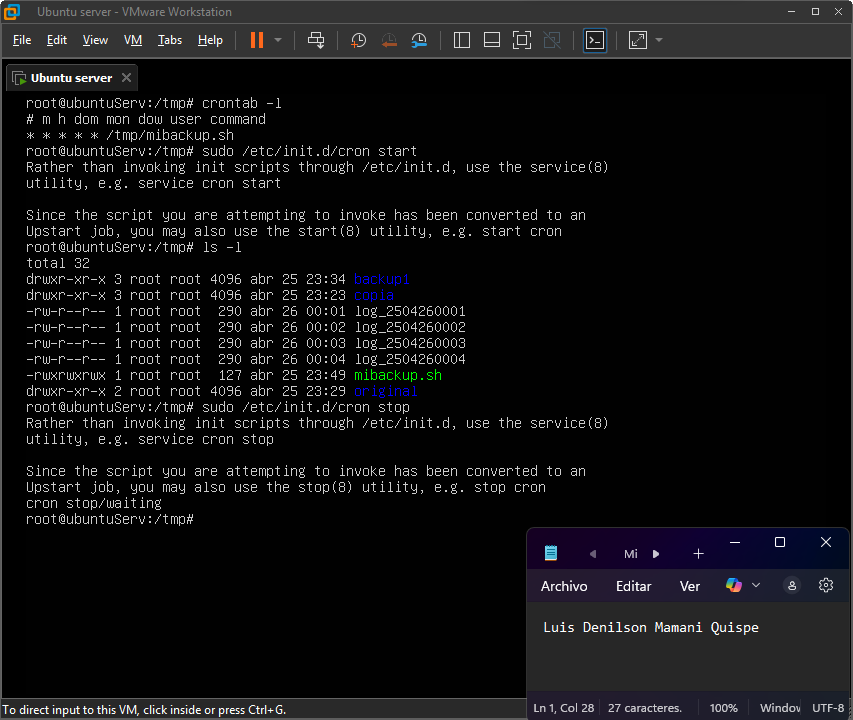
A continuación, iniciamos el servicio:



Si listamos después de cada minuto un nuevo log aparecerá, indicando que la copia se realizó, en este caso cada minuto.

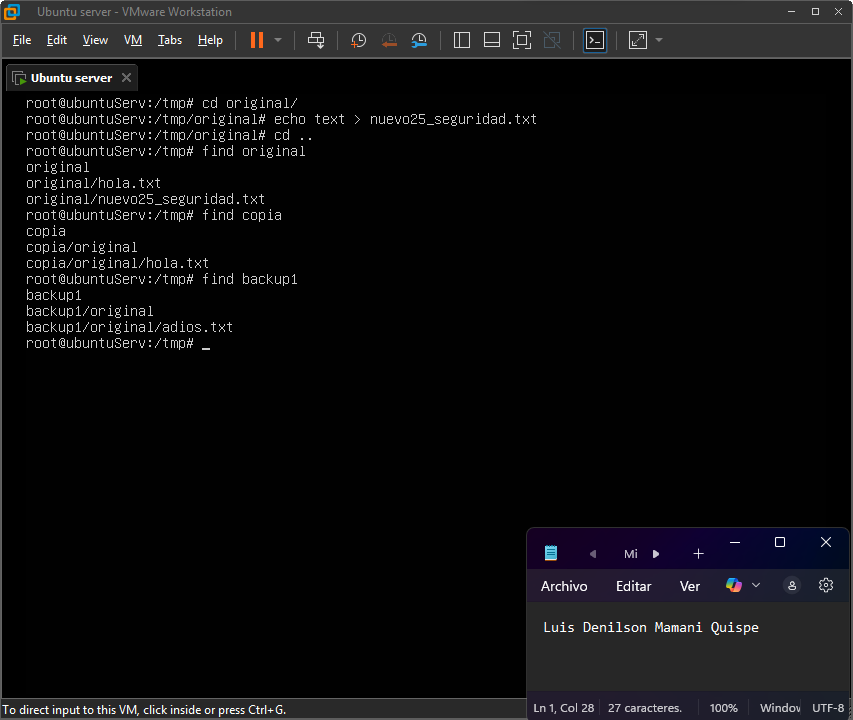


Para detener el servicio utilizamos **cron stop**, caso contrario se llenará el disco de respaldos.



**EVALUACIÓN**

1.- Primeramente, detenga el servicio de cron. Ahora cree un nuevo archivo de texto en la carpeta original con el nombre **nuevo25\_seguridad.txt**, y ayudado por el comando **find** que utilizó en la primera parte, observe el comportamiento de las 3 carpetas.



Comando **find** que utilizó:   
﻿root@ubuntuServ: /tmp# find original

root@ubuntuServ:/tmp# find copia

root@ubuntuServ:/tmp# find backup1

¿Qué archivos están inicialmente en original, copia y backup1?

﻿**original/**

hola.txt

nuevo25\_seguridad.txt

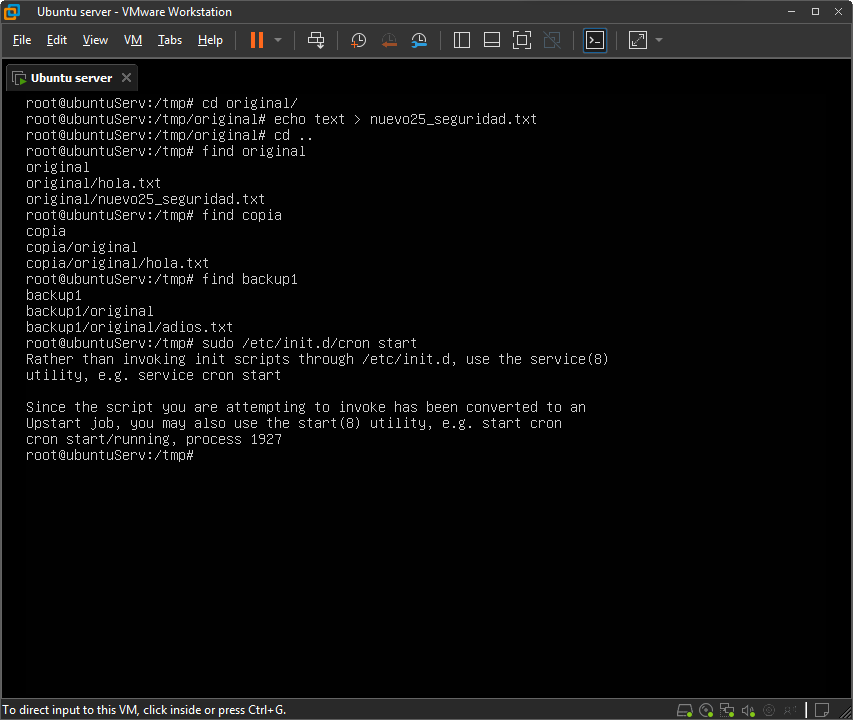
**copia/**

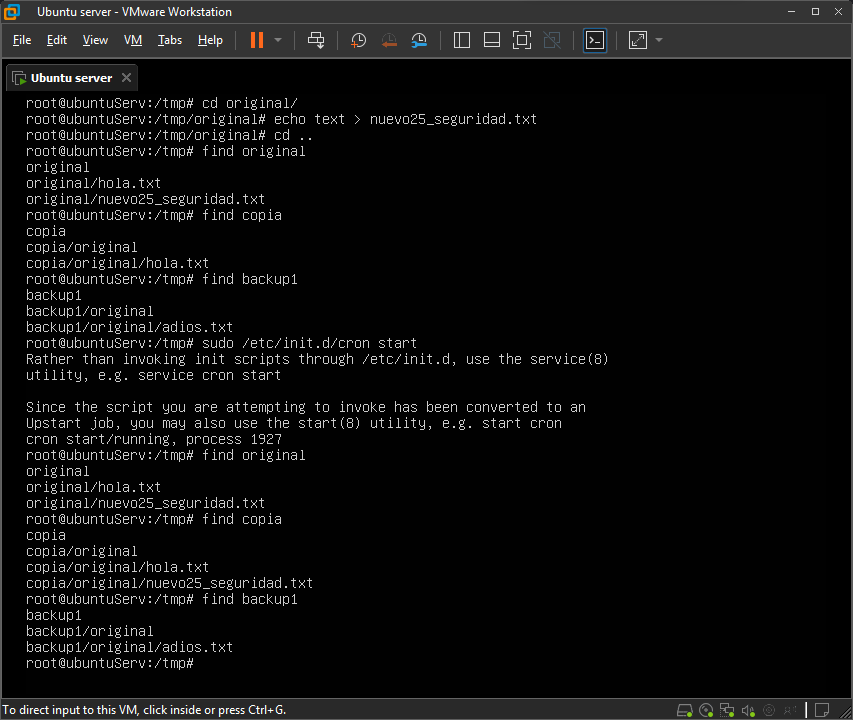
hola.txt

**backup1**

adios.txt

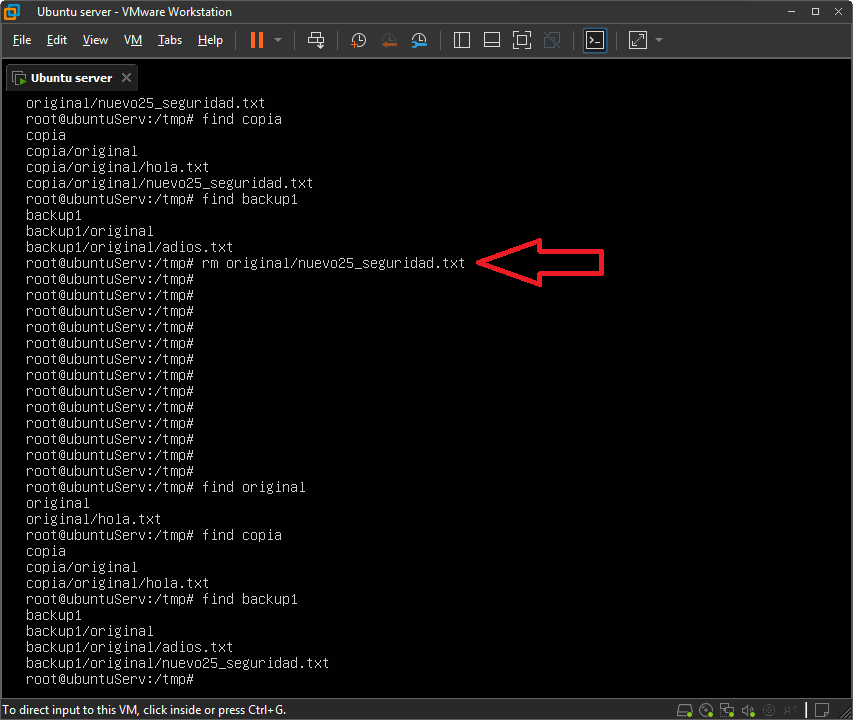
2.- Vuelva a iniciar el servicio de cron, Pasado 1 minuto vuelva a ver las mismas carpetas mediante **find** ¿Qué cambios ocurrieron?





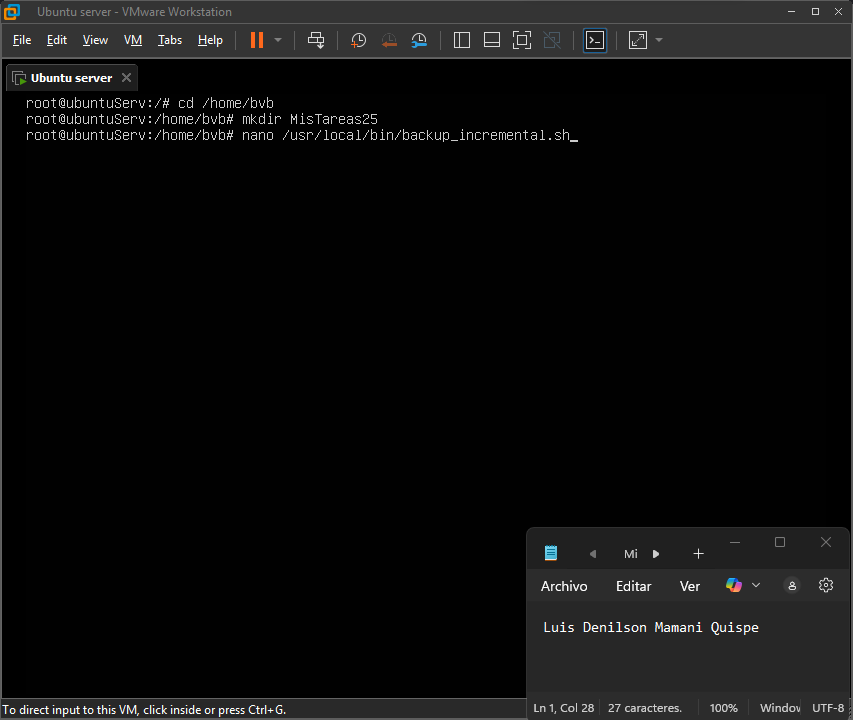
Se hizo una backup del archivo **nuevo25\_seguridad.txt** en **copia/**

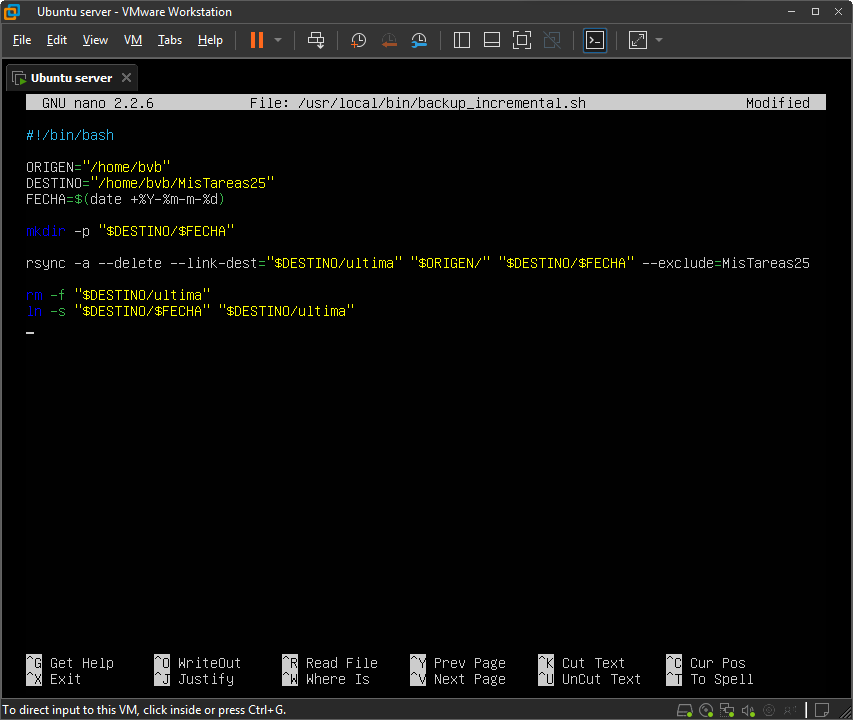
3.- Borre el archivo **nuevo25\_seguridad.txt**, pasado 1 minuto ¿Qué ocurrió? Utilice **find**

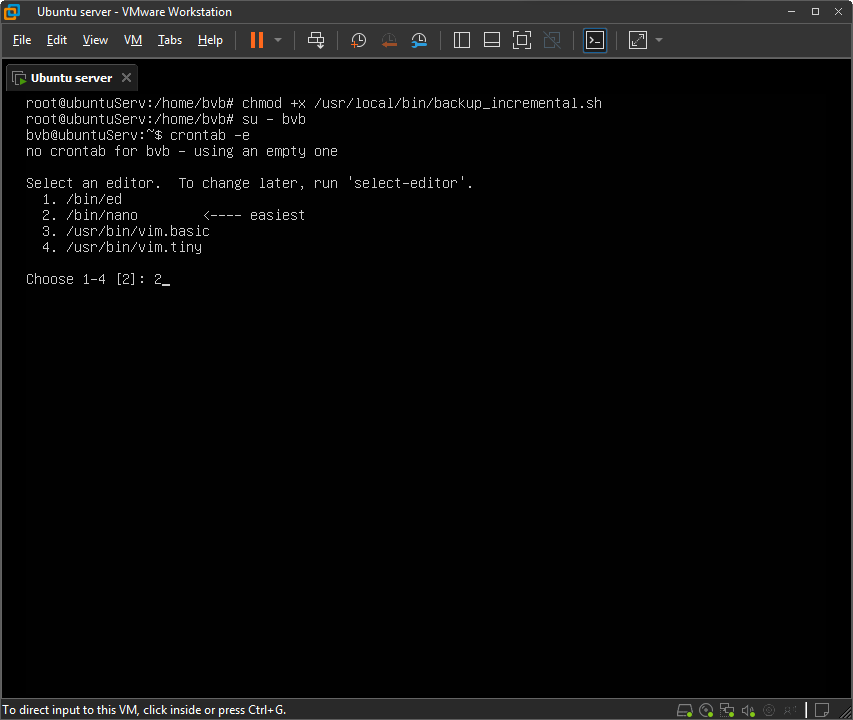


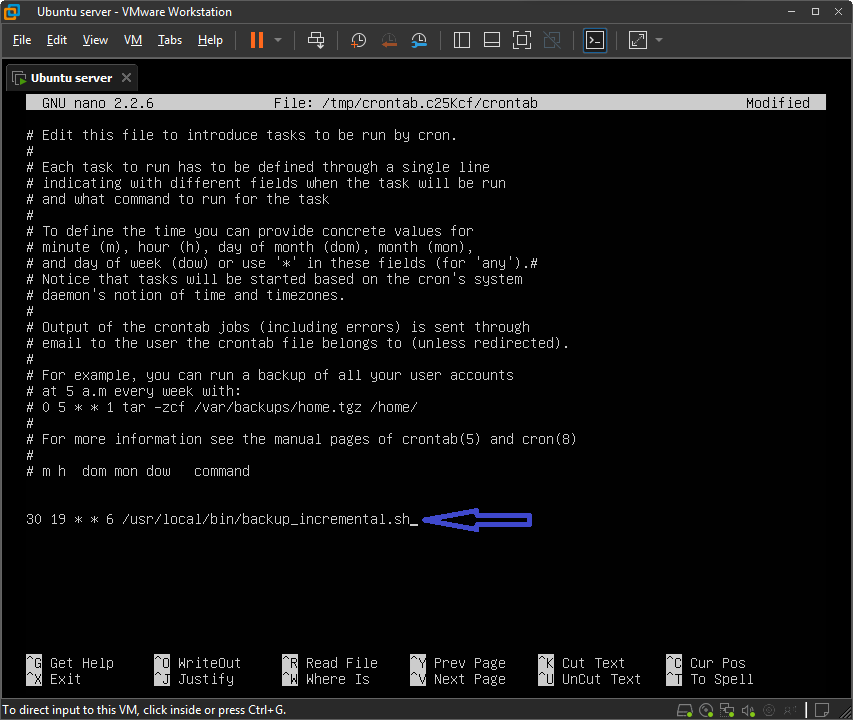
Se elimino el archivo en **original/**, se elimino el backup en **copia/** y al final se hizo un backup del archivo eliminado en **backup1/**

4.- Realice las tareas necesarias para realizar una copia de seguridad incremental a un directorio llamado MisTareas25, para que se realicen automáticamente cada día sábado a las 19:30.





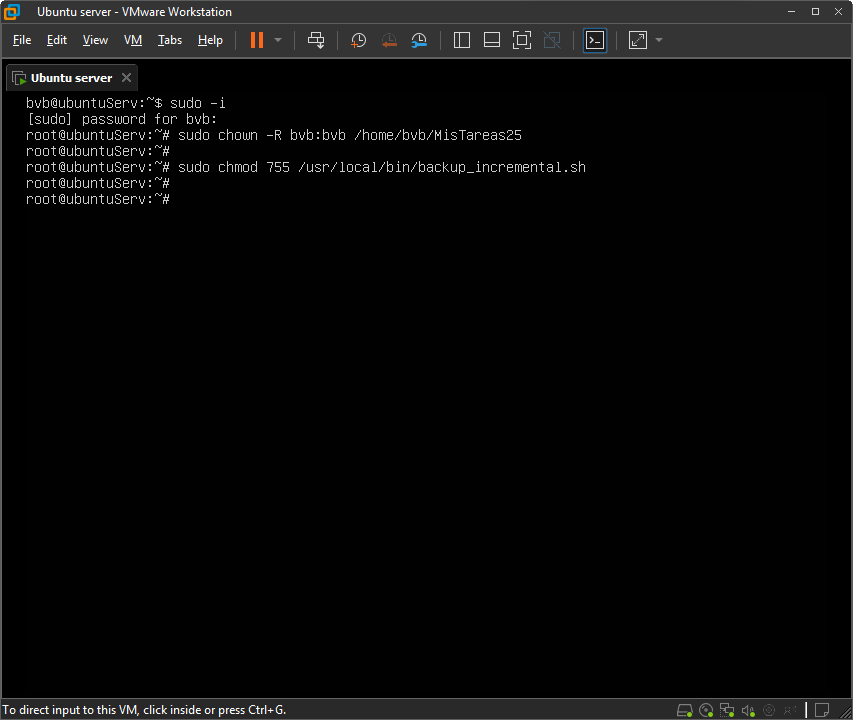


Agregamos al final lo siguiente:  


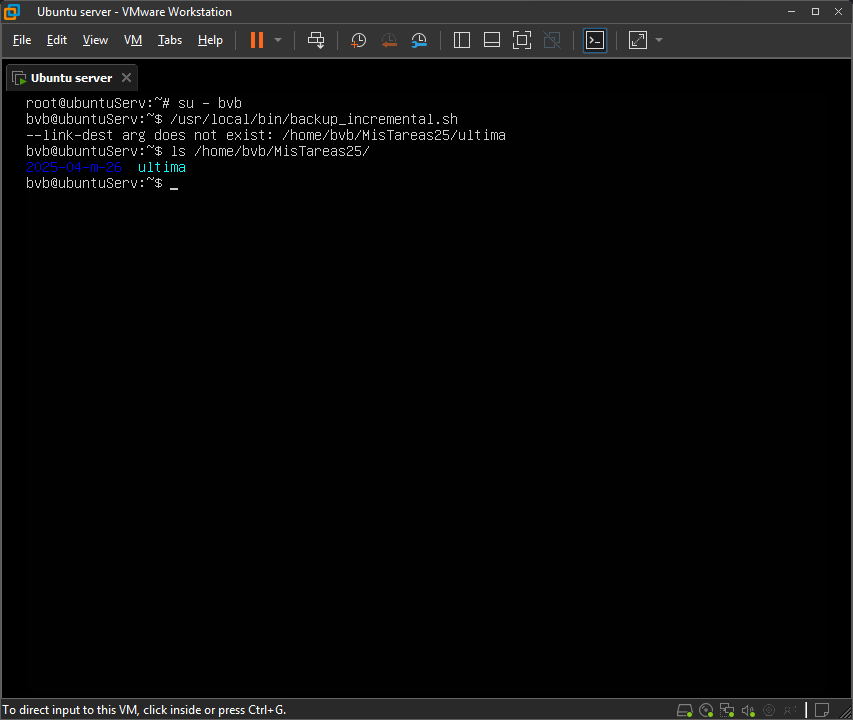
Lo cual programa la ejecución del script todos los sábados a las 19:30 pm

Importante:

Como creamos la carpeta MisTareas25 y el script backup\_incremental.sh con root tenemos que cambiar de dueño MisTareas25 al usuario bvb y dar los permisos correctos al script.



Nos logueamos con el usuario bvb y ejecutamos el script

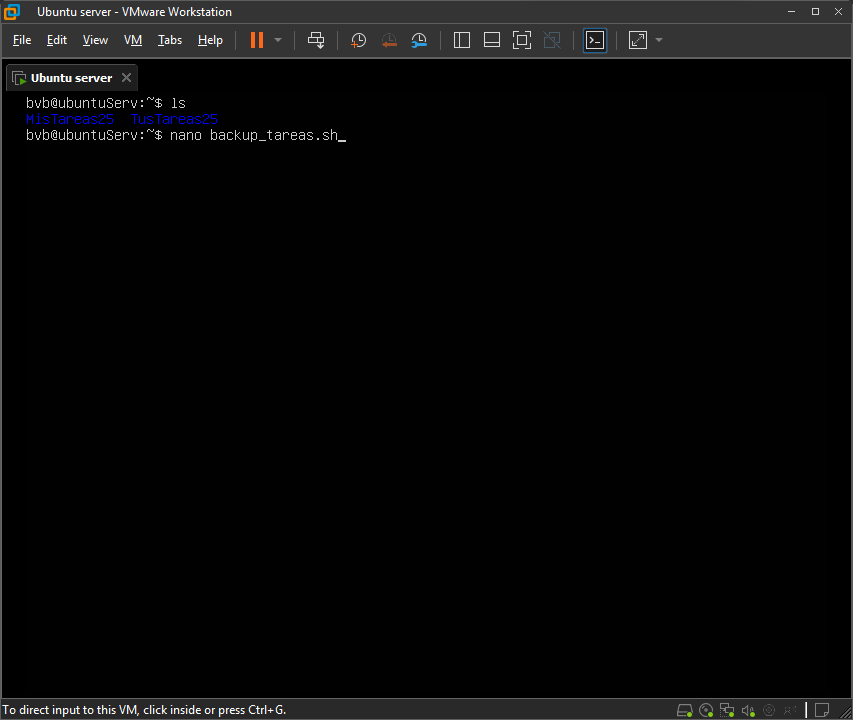


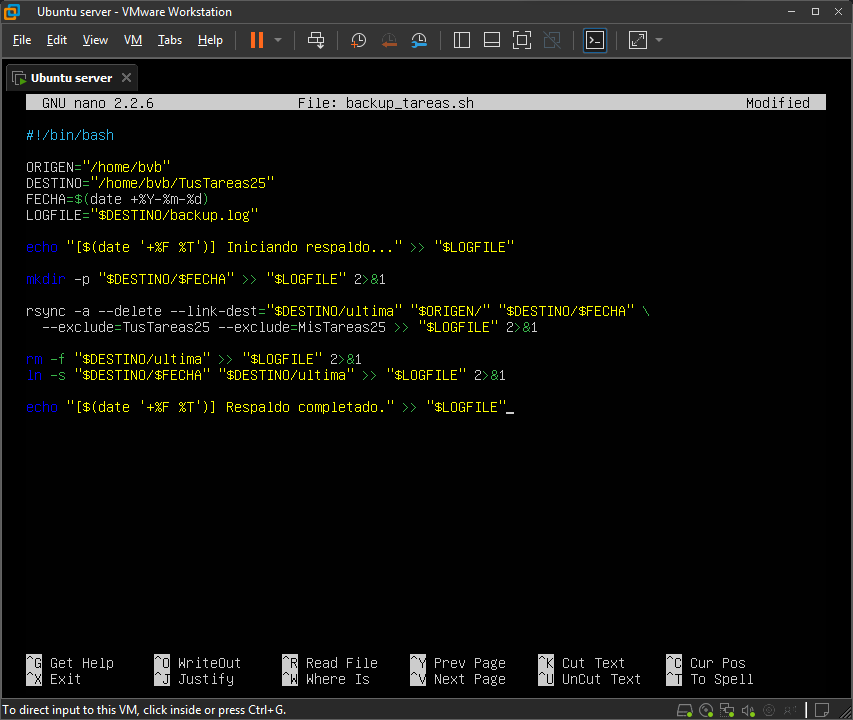
Resultado:

Al ejecutar el script por primera vez, nos mostro el aviso de que la última no existía fue al esperado.

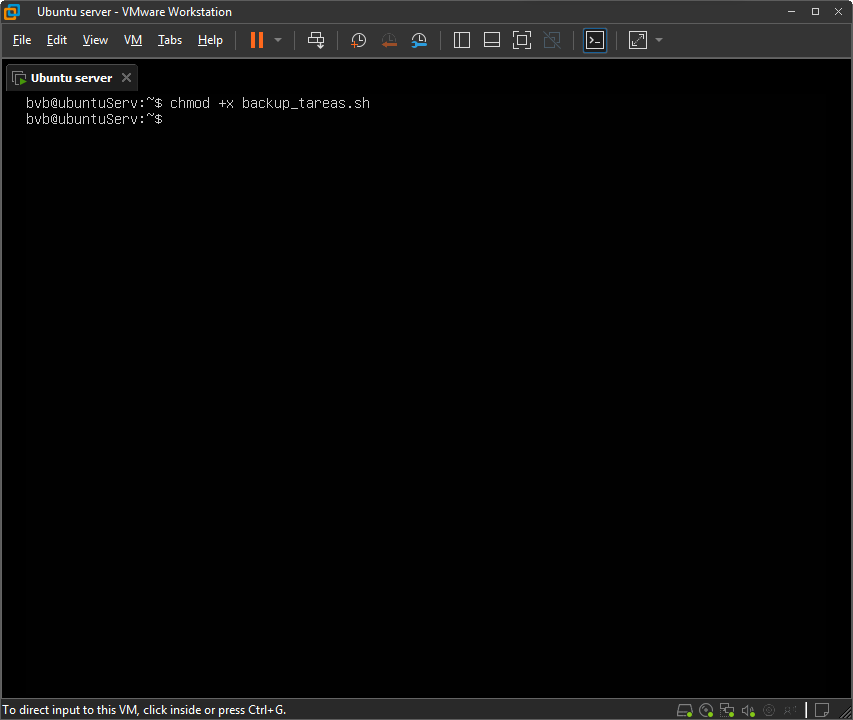
En la foto nos muestra que el script creo una copia completa con la fecha de hoy y estableció el enlace simbólico ultima.

5.- Realice las tareas necesarias para realizar una copia de seguridad incremental a un directorio llamado TusTareas25, para que se realicen automaticaticamente los días lunes, miércoles y viernes a las 00:55 am.

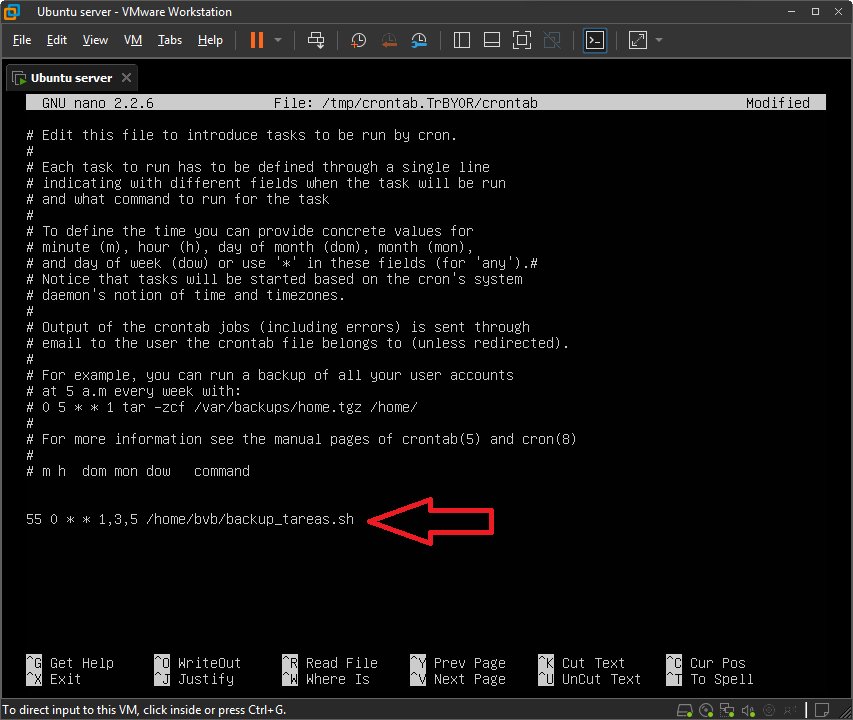


Script:  


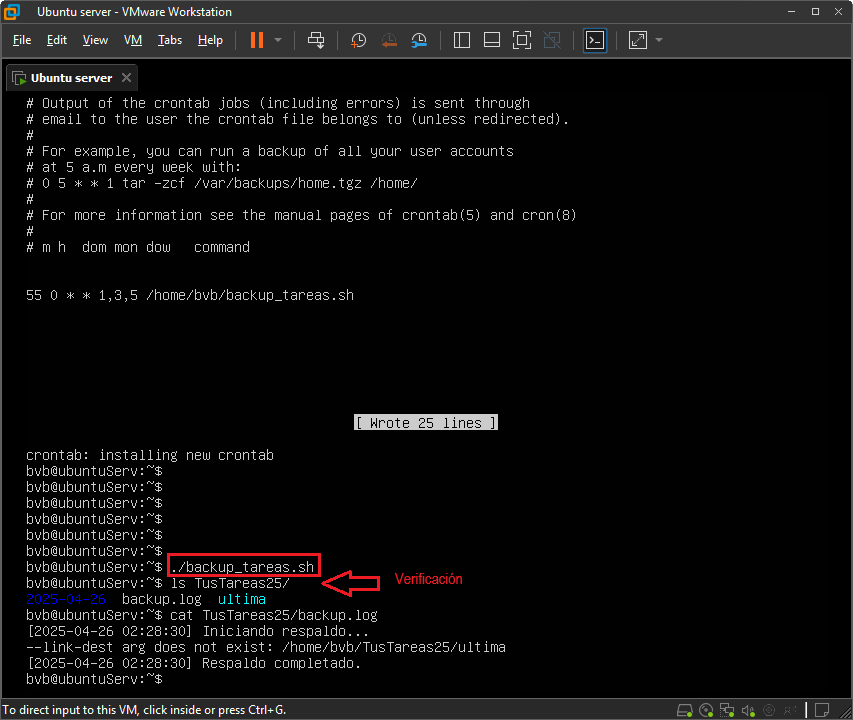
Permisos



Cron



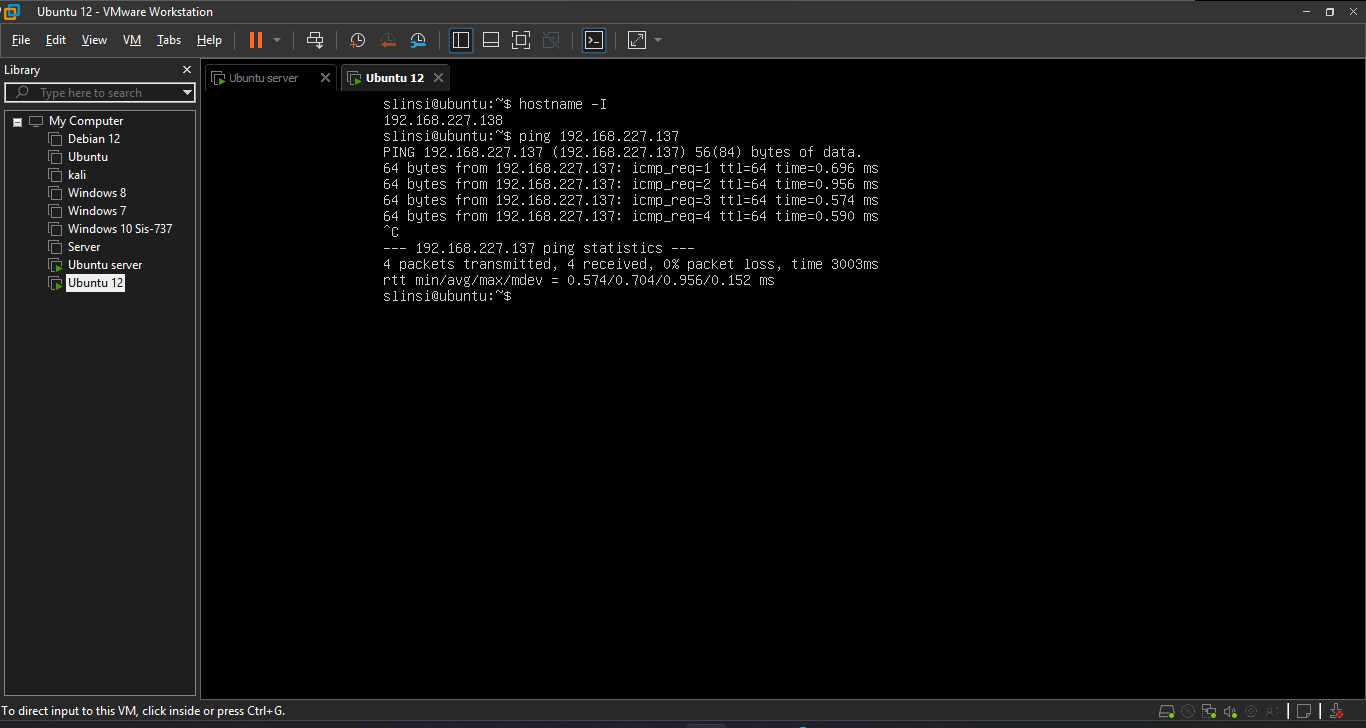
Corremos el script



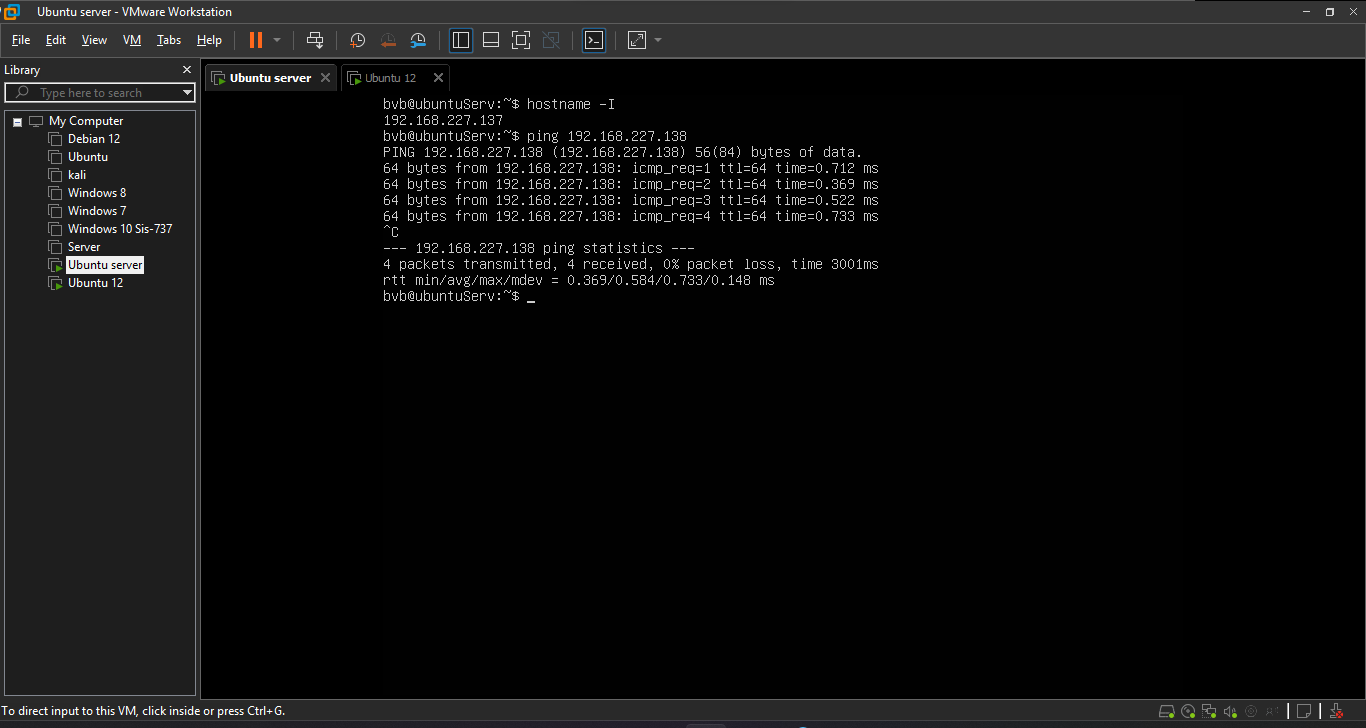
6.- Usando como destino otra máquina virtual, realice una copia programada desde Ubuntu server a la máquina destino mediante red para un día domingo a las 05:20am.

Procedemos a hacer ping entre las maquinas para ver si tienen comunicación

Ubuntu



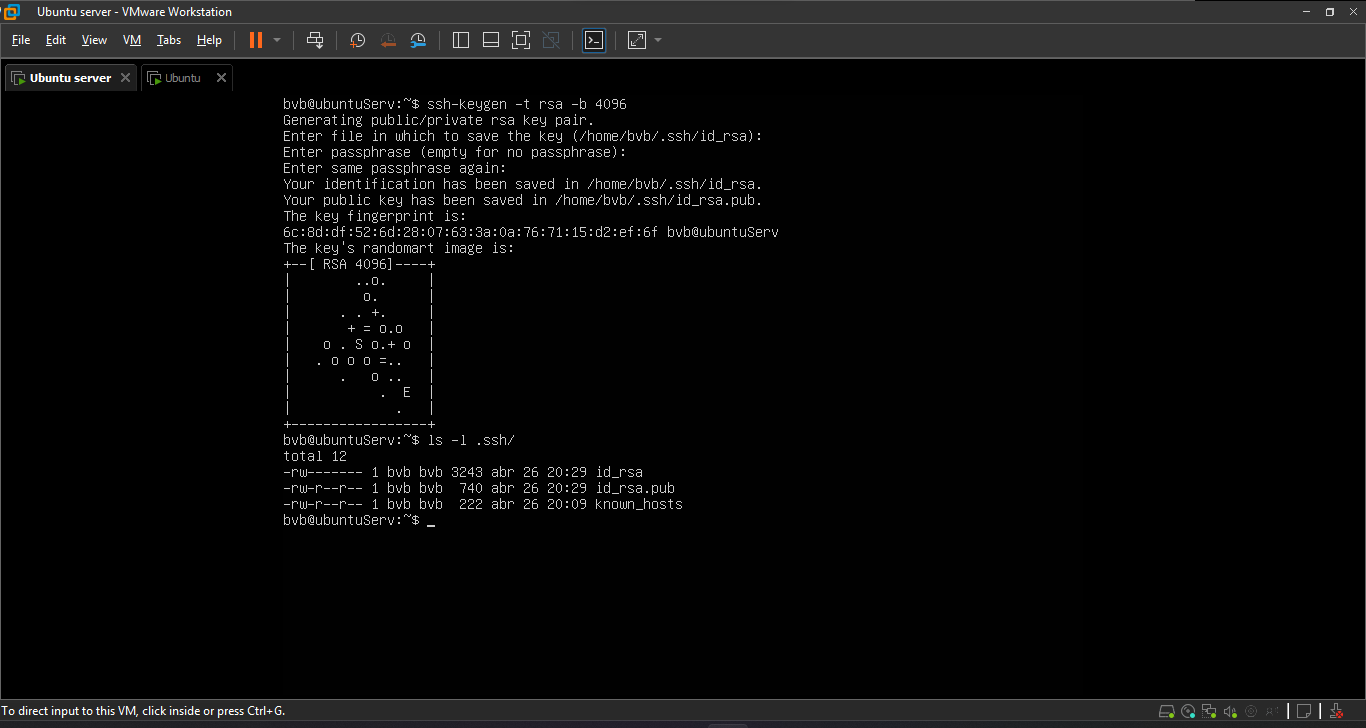
Ubuntu server



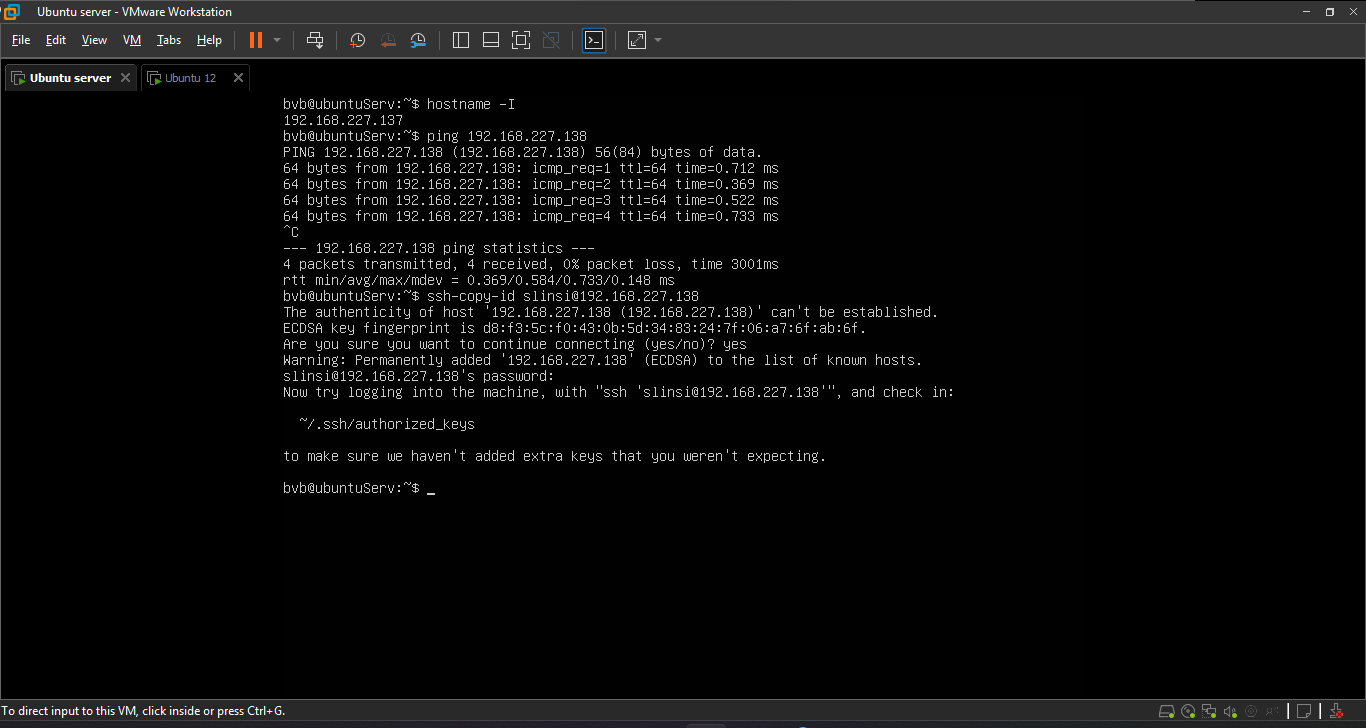
Procedemos con la copia

Paso 1: CONFIGURACION AUTENTICACION SSH sin contraseña

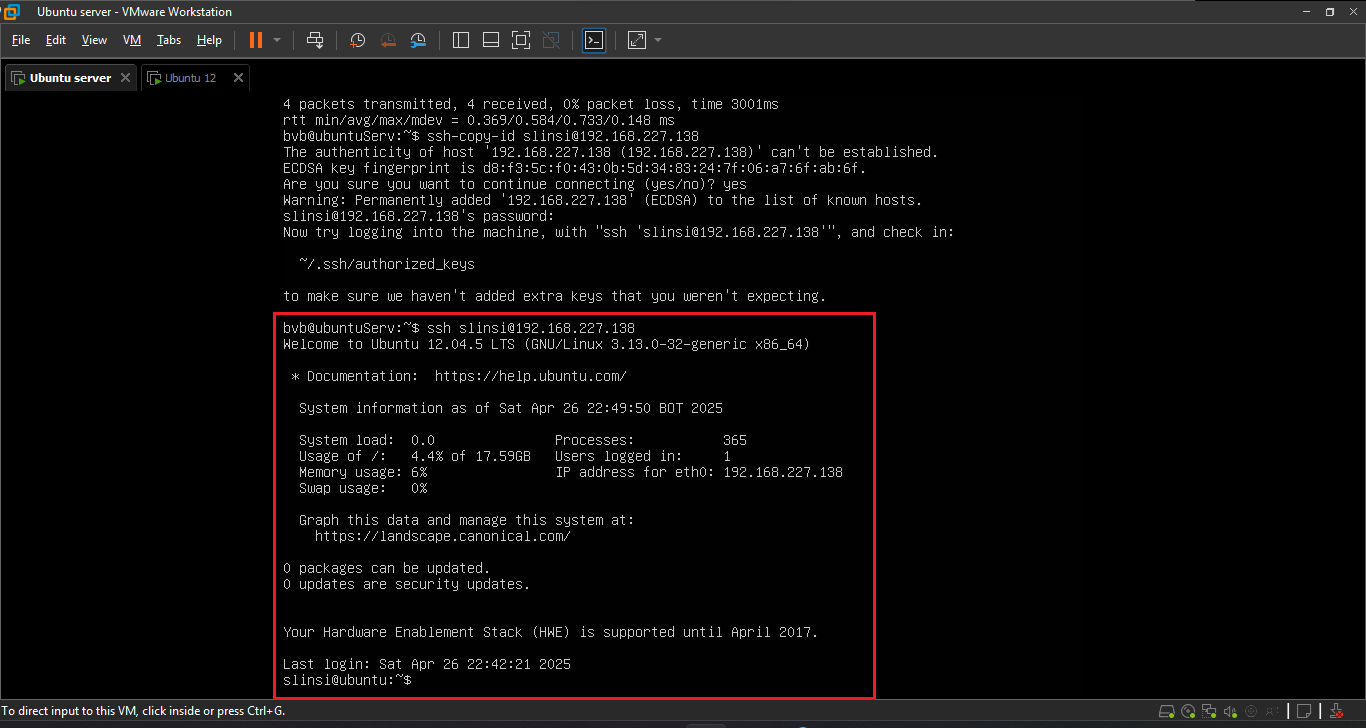
1.- Generar una clave SSH



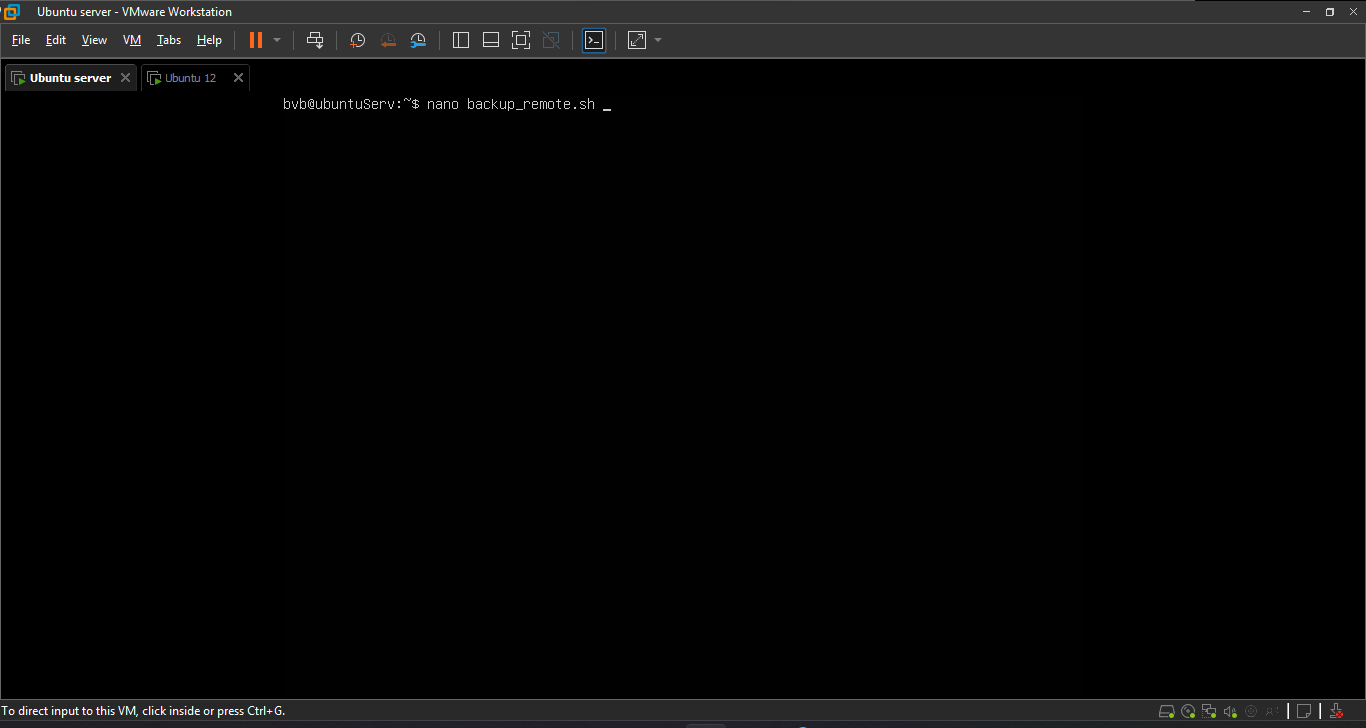
2.- Copiamos la clave publica a la maquina destino, nos pedirá la contraseña



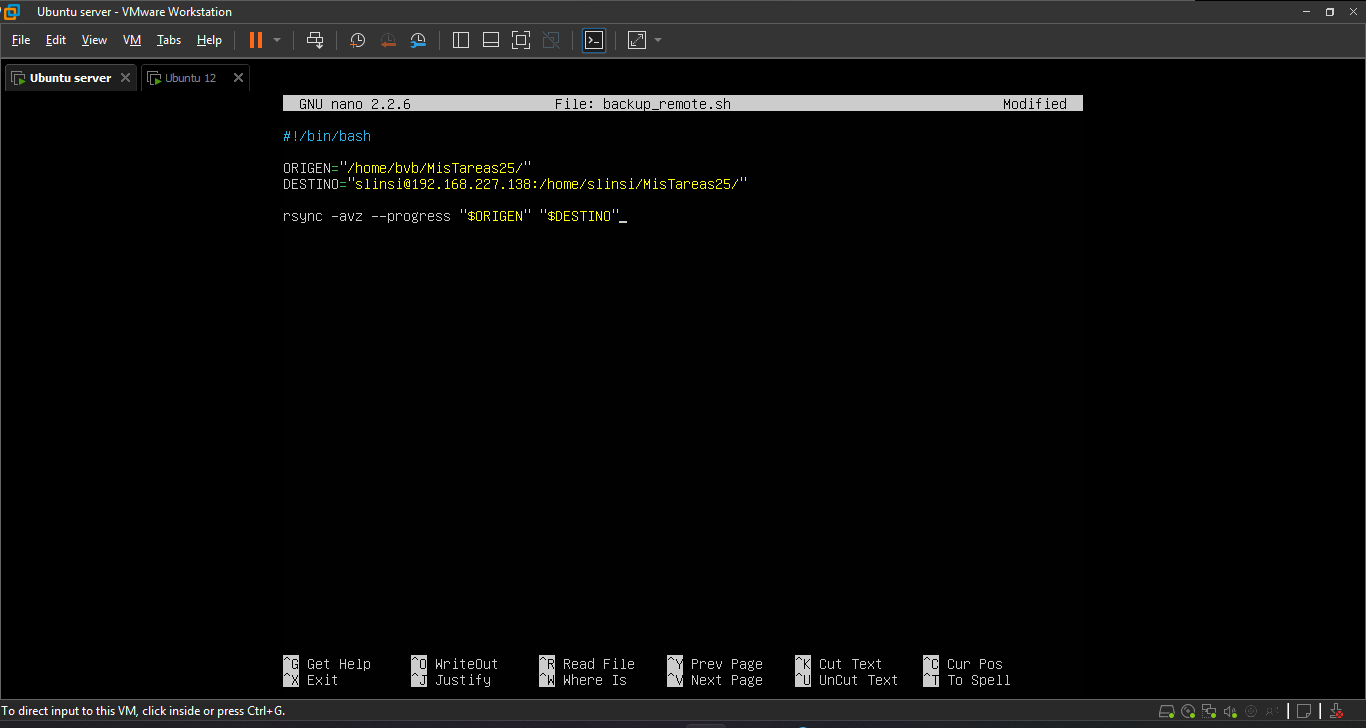
3.- verificamos el acceso sin contraseña



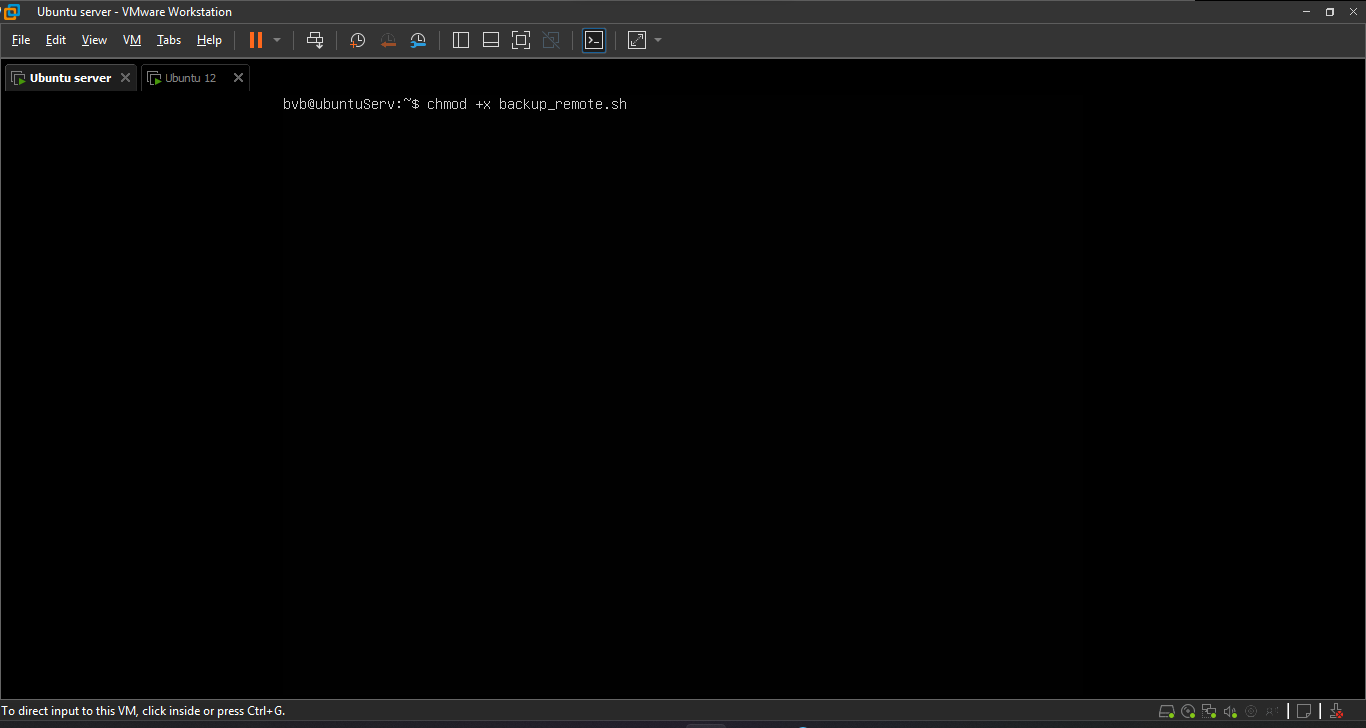
PASO 2: Crear el script de backup



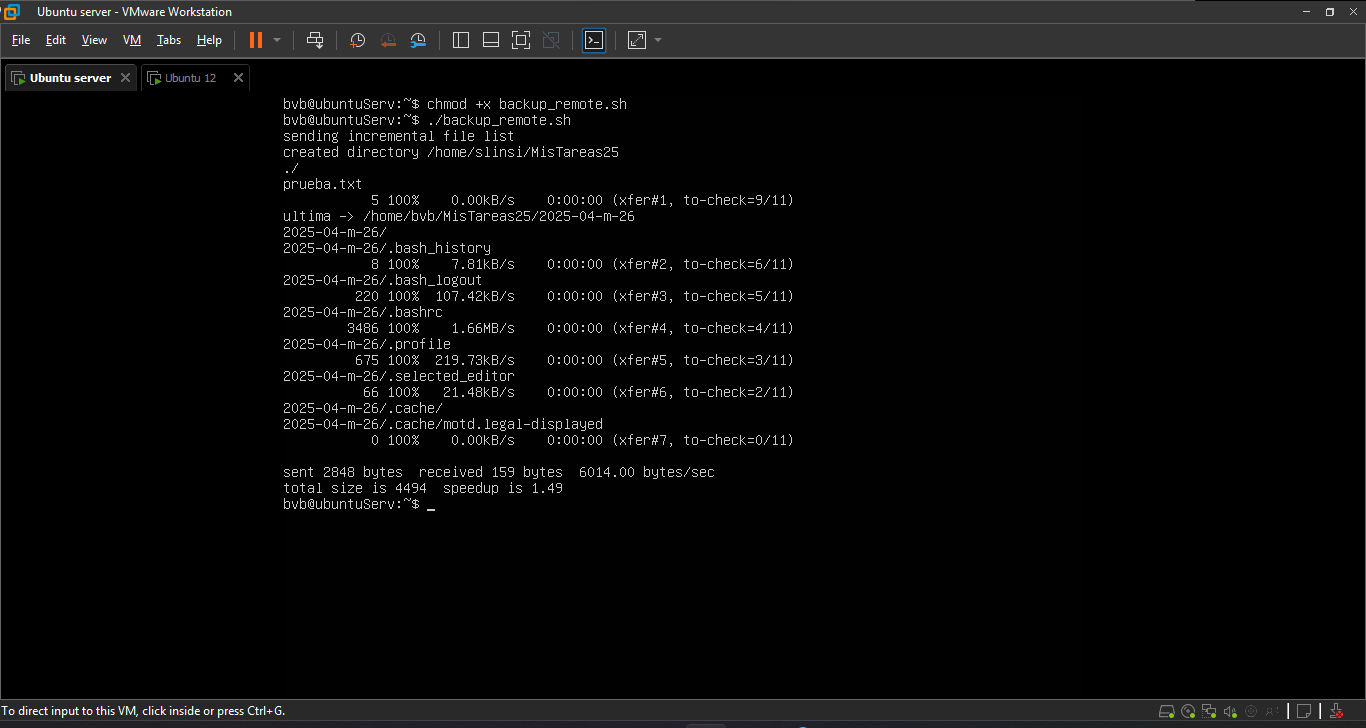
2.- agregar el script

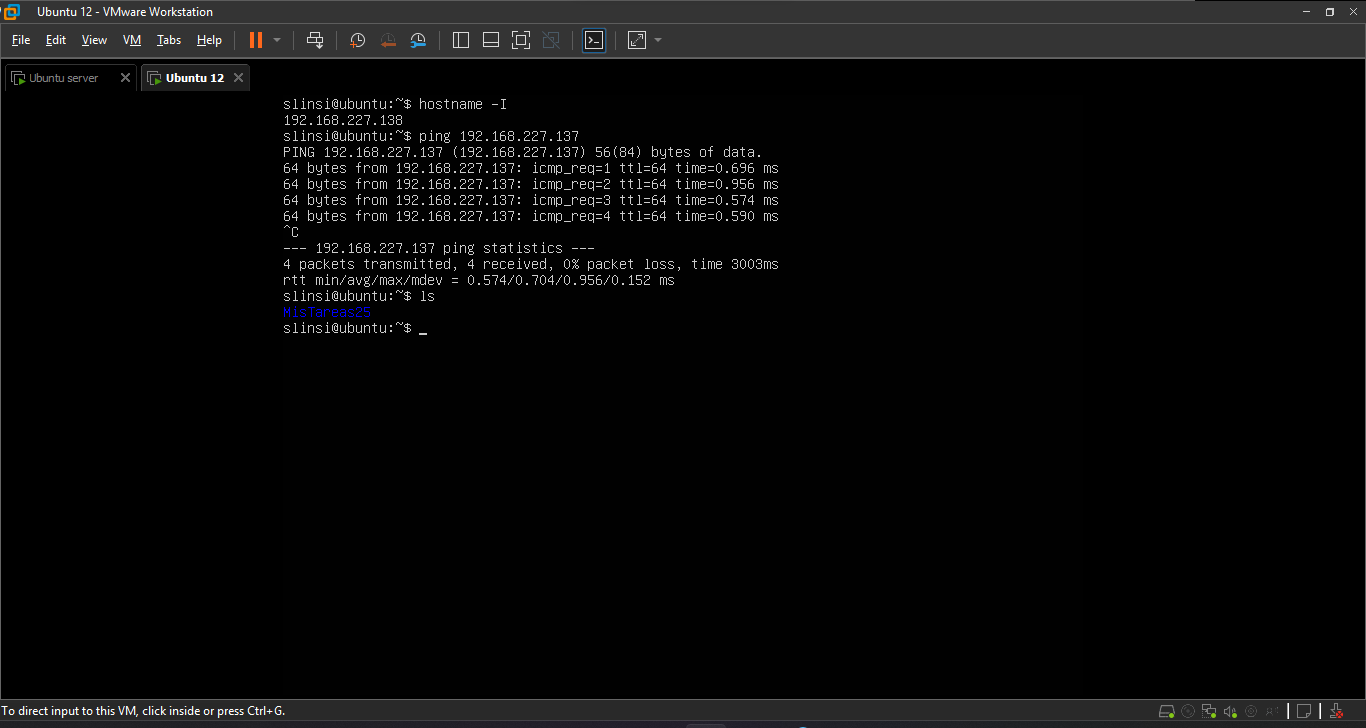


3.- Dar permisos

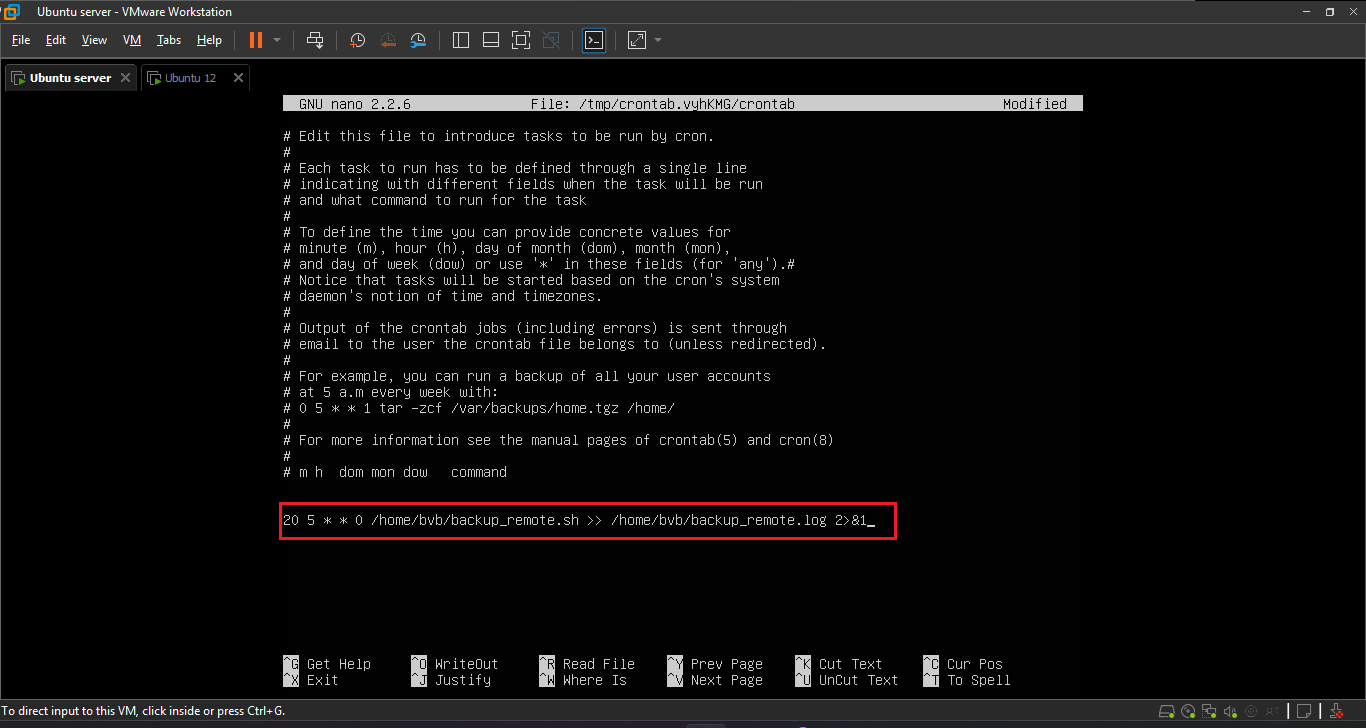


4.- Probar manualmente



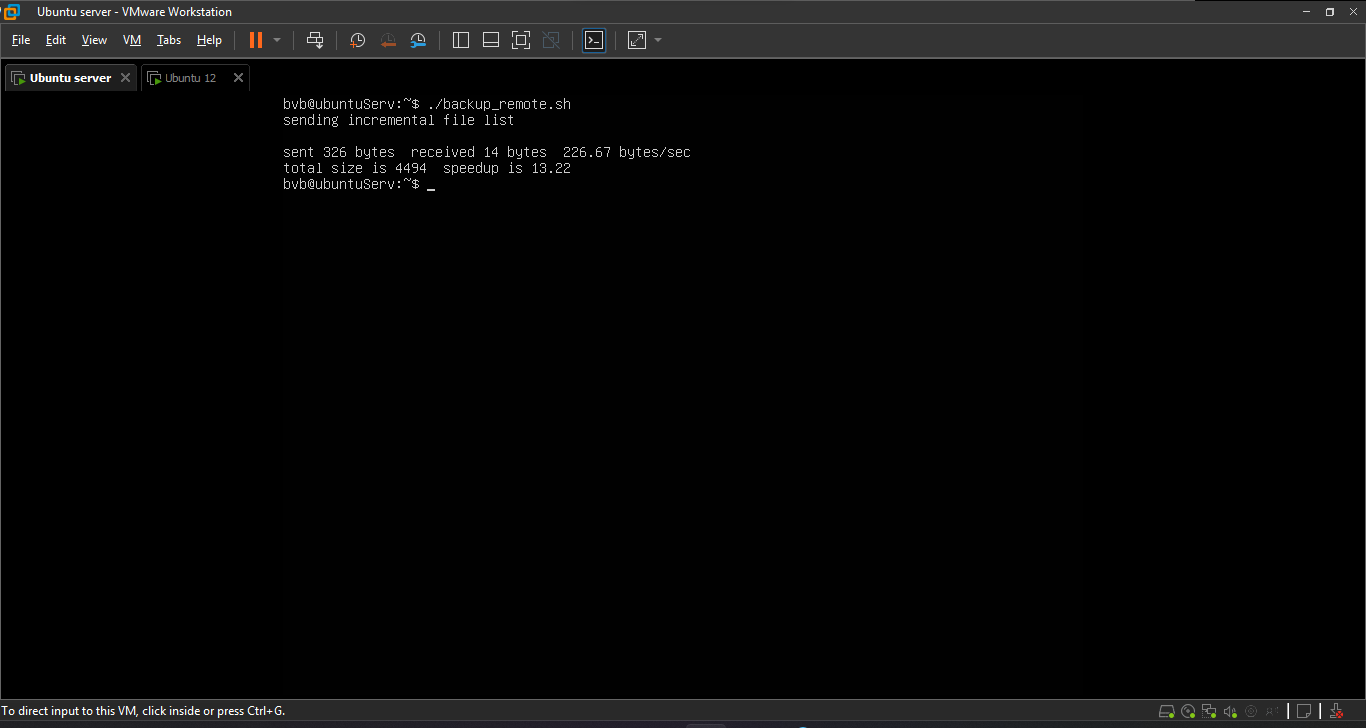


Paso 3: Configurar cron



PASO 3: Probar el archivo de log

1.- Ejecutar el script



Importante: El archivo backup\_remote.log se creara el domingo a las 05:20am