Procedimiento de copias de seguridad.

Adrian Navarro SR2A

Índice

1. Introducción
2. Contenidos
   1. Configuración de Rclone
   2. Control básico de rclone
   3. Funcionalidad del procedimiento
   4. Crontab
   5. Recuperación
3. Introducción

En este documento se detalla la configuración de la herramienta “Rclone” utilizada para la gestión de archivos en sistemas de almacenamiento en la nube.

Clarificación de los scripts de copias de las copias de seguridad, así como su puesta en marcha y automatización con “crontab”

También se incluye un ejemplo sencillo de recuperación de los datos.

1. Contenidos
   1. Configuración de Rclone

Rclone es una herramienta de gestión de archivos en cuentas de almacenamiento en la nube. Es de uso libre y está disponible en las principales distribuciones de Linux.

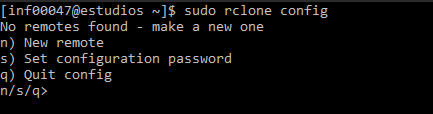
Para instalarla en un sistema Centos:

*sudo yum install rclone*

Para instalarla en un sistema Ubuntu:

*sudo apt install rclone*

Una vez instalado Rclone metemos el comando *sudo rclone config* (los perfiles de rclone se guardan en los home de cada usuario, por lo que si queremos que nuestro usuario tenga también acceso a la cuenta de almacenamiento tendremos que repetir el proceso de configuración, pero sin el “sudo” *rclone config*).

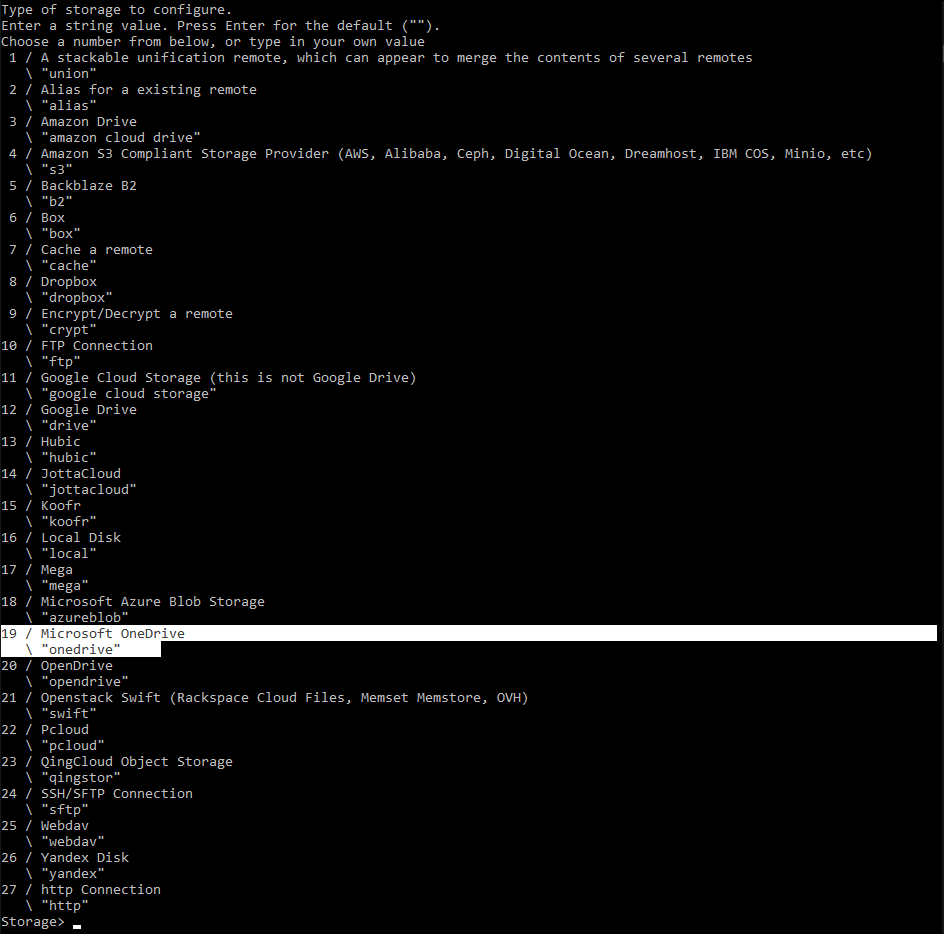


Como se puedo ver no tenemos ningún “remote” guardado aun por lo que pondremos “n“ new remote y le daremos el nombre que deseemos. En mi caso he escogido cloud ya que así se hace referencia en los scripts.



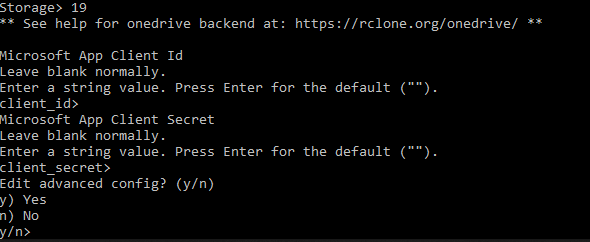
Se nos desplegara la lista de los servicios en la nube a los que tiene acceso rclone. Buscamos el equivalente al que nosotros deseamos introducimos el número que le hace referencia y continuamos.

Se yo he escogido OneDrive de Microsoft, los procesos de configuración pueden variar según la plataforma.



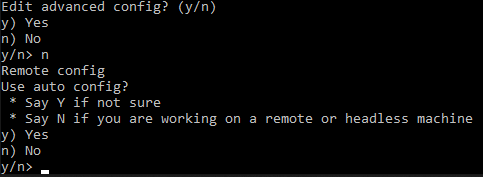
A continuación, nos aparecerá dos apartados en los cuales podemos darle a entre y tomar los valores por defecto.

Sin embargo, también nos saldrá un mensaje de si queremos editar la configuración avanzada. Los valores por defecto son los que se han utilizado y no hay problemas por lo que le indicamos “n” no.

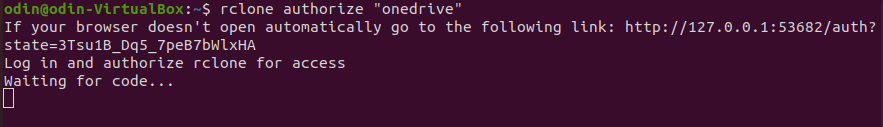


El siguiente mensaje nos pregunta se estamos trabajando con una maquina con interfaz gráfica o si estamos trabajando con una maquina sin interfaz gráfica (o por ssh)

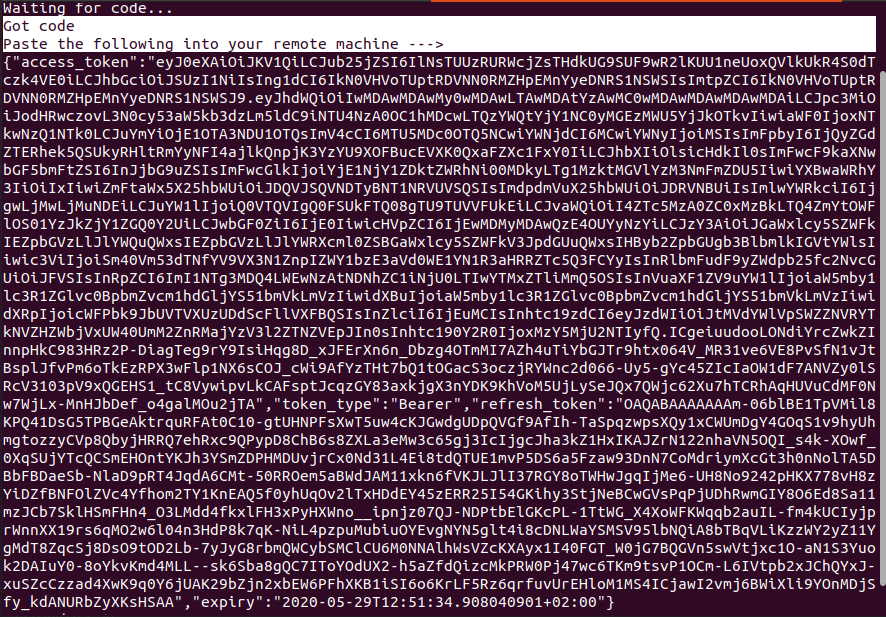
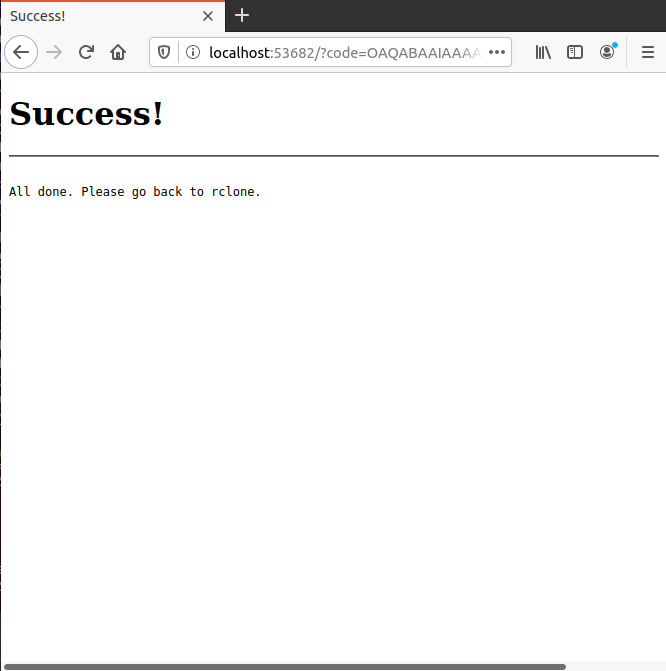
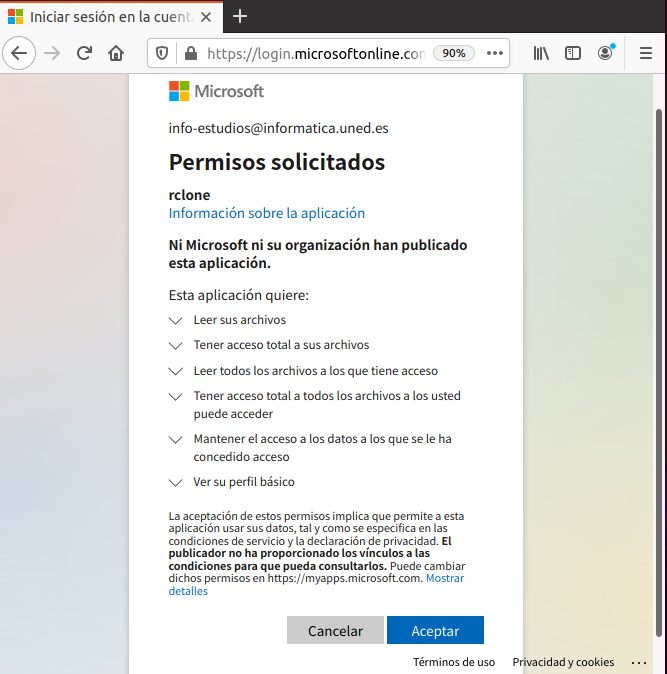
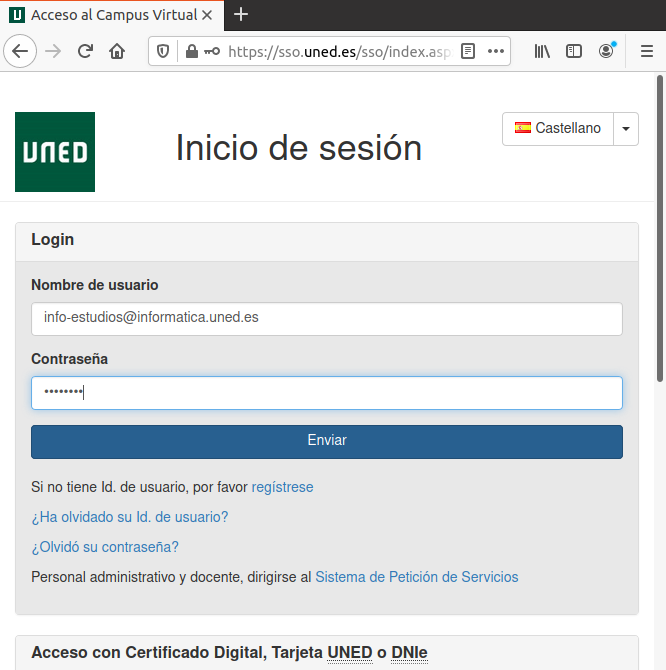
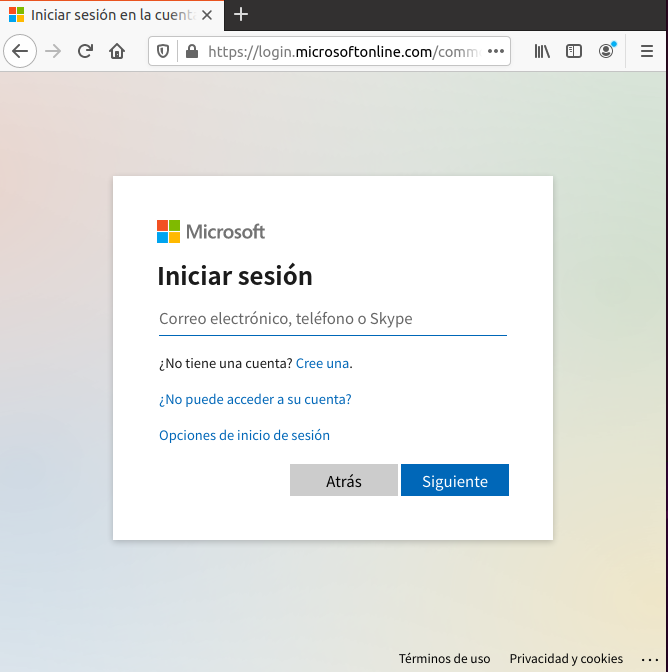
Si tenemos interfaz ponemos “y” y si no tenemos ponemos “n”. En mi caso no tenía. (El proceso es el mismo pero con un pasito intermedio)



En otra máquina Linux en la que tengamos acceso a la interfaz grafica, instalamos rclone y metemos el comando rclone authorize "onedrive”



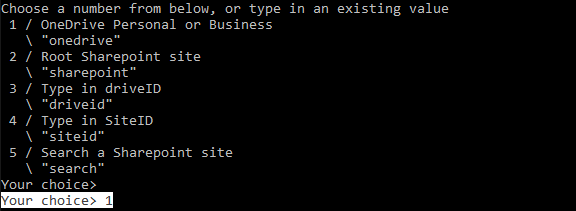
Esto debiera abrirnos el navegador en la con la dirección del portal de inicio de sesión de Microsoft. Nos validamos y volvemos a la terminal.



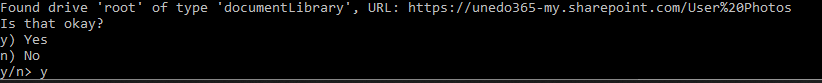
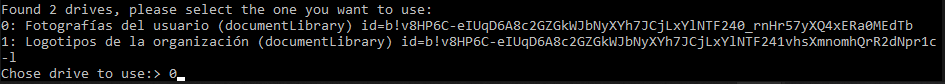
Hacemos como se nos pide y copiamos todo el texto que está entre las llaves y lo pegamos en la terminal de la maquina en la que estamos configurando el rclone



Continuamos u nos preguntara que tipo de cuanta de almacenamiento es. Diremos que es Onedrive personal o business “1”



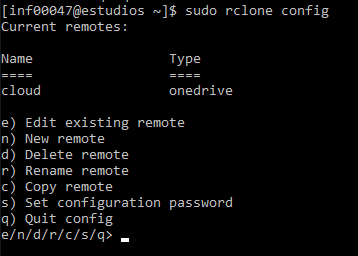
Ahora seleccionamos el disco raíz al que equivaldrá el remoto, en mi caso era la primera opción “0”



Confirmamos que todo este correcto y le damos a “y”



Y ya tenemos nuestro remoto configurado



* 1. Control básico de rclone

Rclone tiene sus propios comandos para funcionar, pero son bástate sencillos. Las rutas de los remotos son nombreDeRemoto:directorio1/directorio2/

Si queremos copiar algo de nuestra maquina a la nube

rclone copy <ruta\_local> <ruta\_en\_la\_nube>

Ejemplo:

rclone copy /home/odin/test.txt cloud:home/odin/

Y si queremos copiar algo de la nube a nuestra maquina el proceso es el mismo, solo hay que invertir las rutas.

Supongamos que queremos eliminar algún elemento de la nube:

rclone delete <ruta\_de\_la\_nube>

Se queremos ver el contenido de un directorio en la nube

rclone ls <ruta\_de\_la nube>

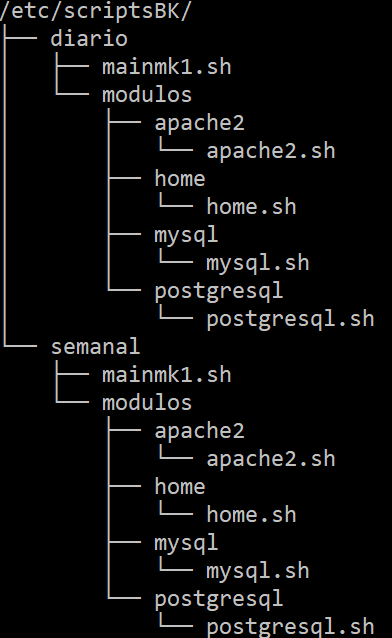
Luego hay un comando un poco contra intuitivo

rclone sync <ruta1> <ruta2>

Este comando te sincroniza el contenido de la ruta1 en la ruta2, pero si en la ruta 2 hay algún dato, también lo elimina. De tal manera que realmente te crea un espejo de una ruta en la otra.

* 1. Funcionalidad del procedimiento

El paquete de scripts cuenta con dos vertientes, la semanal y la diaria. A nivel de estructura son los dos iguales con la única salvedad de que la semanal cuenta con un bloque de código que se asegura de que haya x número de archivos y que su nombre incluye la fecha en la que se realizó la copia. La estructura y filosofía de trabajo es la siguiente:



Como podemos ver se cuenta con un archivo “main.sh” que es el que se ejecutará y luego en el directorio “modulos” se incluirán los distintos servicios de los que se quieren hacer las copias de seguridad.

En “main.sh” contamos con la inclusión previamente comentada de los distintos módulos. Por otro lado, se establecen las variables de la fecha en la que se ejecuta el procedimiento, creación de la carpeta “/backups/”, la creación del archivo de logs y la subida a la nube del propio archivo de logs.

El procedimiento está diseñado de tal manera que si se quiere introducir un nuevo servicio sea tan fácil como clonar uno de los módulos ya existentes y modificar un par de variables.

Cada módulo te creara uno o dos directorios “conf” y/o “data”, dentro de ellos encontraremos paquetes comprimidos. Dependiendo, encontraremos archivos de datos o de configuración de los servicios. Estos se habrán cogido de un directorio temporal ubicado en el directorio “/tmp/” que a su vez habrán sido copiados ahí automáticamente por el script. De tal manera que si se necesita añadir algún otro archivo de simplemente con añadir un *cp –rp <archivoA\_Añadir>* en el script estaría listo.

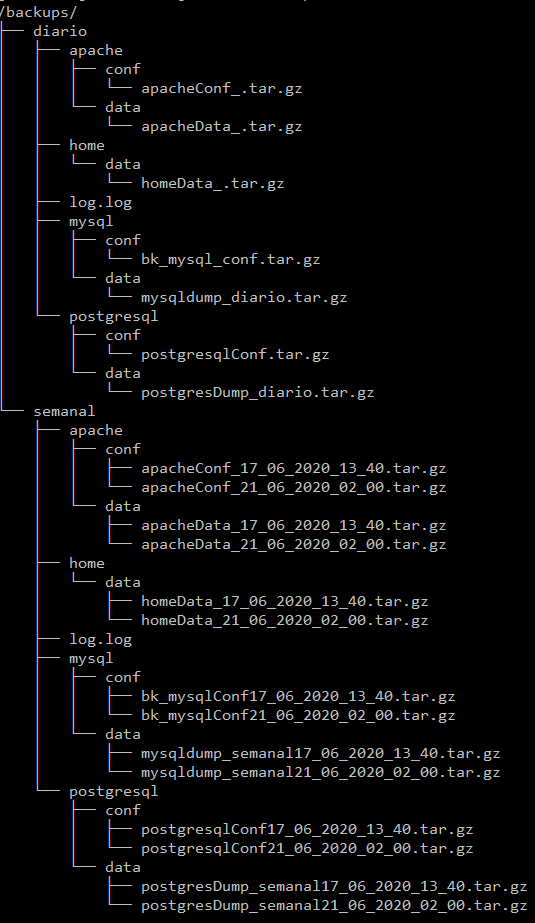
En algunos casos en los directorios de “data” no bastara con copiar y pegar, ya que por ejemplo con las bases de datos hemos tenido que volcarlas utilizando sus respectivos dump.

Un dato importante a tener en cuenta es que por defecto se comprimirán los paquetes con una extensión .gz.

Continuando con el procedimiento encontramos una estructura de control que nos evaluara el tamaño del paquete y si es mayor de 10GB no los dividirá en partes más pequeños. Esto es por una limitación de la nube de Microsoft que no nos permite subir archivos de más de 10GB.

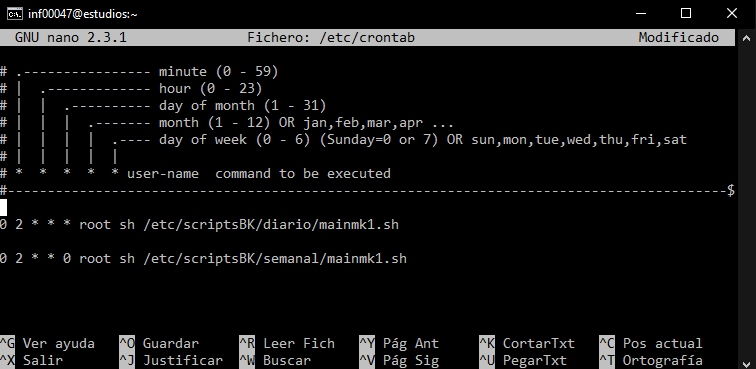
En los módulos de la vertiente semanal nos encontraremos un último bloque de código que se comprueba el número de archivos en el directorio y si es mayor a 3, eliminara el más antiguo dejando solamente 2 archivos.

Por supuesto el procedimiento te genera una copia de seguridad en el directorio /backups/ y luego también te genera la misma estructura en la nube.



* 1. Crontab

Crontab es una herramienta que tiene Linux que nos permite automatizar tareas. Nosotros establecemos cuando han de ejecutarse las tareas. Para editarlo tecleamos *sudo nano /etc/crontab*



Si nos fijamos, nos dice que el primer campo hace alusión a los minutos, el segundo campo a la hora, el tercero al día del mes, el cuarto al mes y el quinto al día de la semana. Luego está el usuario con el que se ejecutara la tarea y la propia tarea que queremos que realice.

En la imagen vemos que tengo configurado que me ejecute el main.sh diario todos los días de la semana de cada mes a las 02:00 con el usuario root.

*0 2 \* \* \* root sh /etc/scriptsBK/diario/mainmk1.sh*

También tengo configurado que me ejecute el main.sh semanal solo el domingo de todas las semanas de todos los meses a las 02:00 como root.

*0 2 \* \* \* root sh /etc/scriptsBK/diario/mainmk1.sh*

* 1. Recuperación

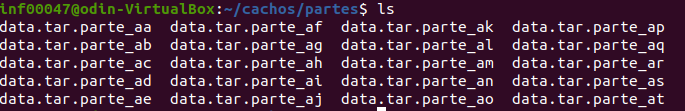
Dependiendo del servicio, el proceso de recuperación puede variar ligeramente.

Lo primero es ser conscientes de en qué punto nos encontramos. El peor de los casos sería un fallo catastrófico de la máquina que requiriese instalar desde cero.

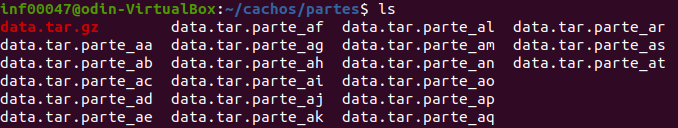
En este caso lo primero que tendríamos que hacer sería instalar y configurar de nuevo rclone para tener acceso a nuestras copias de seguridad alojadas en la nube y descargarlas. Si a la hora de descargar vemos que lo que se nos descarga es una carpeta, esto se debe a que el paquete pesaría más de 10 GB y se tuvo que dividir.

Para solventar este problema tendremos que juntar las partes. No metemos en la carpeta y ejecutamos

*cat directorio\_con\_las\_partes/parte\_\* > directorio\_con\_las\_partes/data.tar.gz*

**

**

**

Posteriormente tendríamos que desempaquetar los archivos descargados.

*tar –xvf <paquete> -C <directorio\_de\_destino>*

Este comando nos desempaquetaría el archivo en el directorio que queramos.

El siguiente paso sería instalar los servicios que necesitemos, una vez instalados, paramos el servicio y remplazamos los archivos de configuración creados por los que teníamos guardados de nuestras copias de seguridad y reiniciar el servicio. Hacemos lo mismo con los archivos de datos y si no hay conflicto de permisos ni nada por el estilo ya estaría restaurado. (en caso de las bases de datos tendríamos que insertar los archivos .sql)