ASR – Trabajo de Teoría

Pablo José Pérez Díez

UO282440

1. Introducción

En este trabajo se aborda el tema de las copias de seguridad (backups) y los sistemas de almacenamiento diseñados para resistir fallos. La importancia de los backups radica en su capacidad para garantizar la integridad y disponibilidad de los datos ante posibles pérdidas ocasionadas por fallos técnicos, errores humanos o ataques maliciosos.

El proyecto se divide en dos partes principales. En la primera, se implementa un sistema de creación de copias de seguridad automatizado, que genera backups de manera eficiente y los transfiere a un servidor externo para su almacenamiento seguro. Para ello, se ha utilizado Rsync para la generación de las copias de seguridad, OpenSSH y scp para su transferencia, y un script que permite la ejecución automatizada de todo el proceso a través de Cron en un entorno Linux.

En la segunda parte, se exploran herramientas más avanzadas y profesionales para la gestión de copias de seguridad. Se empleará Restic, una herramienta moderna y eficiente que permite realizar copias de seguridad cifradas, incrementales y con deduplicación de datos, simplificando el proceso en comparación con Rsync. Además, se estudiarán otras soluciones como Borg, con el objetivo de analizar sus ventajas y desventajas en distintos escenarios de uso. Finalmente, se compararán los enfoques utilizados, evaluando la seguridad, eficiencia y facilidad de implementación de cada método.

Este trabajo tiene como objetivo desarrollar sistemas funcionales y automatizados de backups usando diferentes métodos y herramientas, también se busca ofrecer un análisis comparativo de distintas herramientas y estrategias, proporcionando una visión más completa sobre las mejores prácticas en cuanto a Backups.

2. Rsync

En esta primera parte del trabajo se utilizarán las máquinas virtuales de Alma Linux y Windows Server 2022. El ejercicio consiste en crear un sistema de generación de copias de seguridad periódico y automatizado en la máquina Linux y que estas copias sean enviadas al Windows Server 2022 para su almacenamiento.

En comparación al segundo ejercicio del trabajo, aquí se realizará el sistema de copias de seguridad de una manera más manual y rudimentaria, mediante la creación de un script que automatice todo el proceso.

Para este ejercicio se ha utilizado Rsync (Remote Sync), una herramienta de línea de comandos utilizada para la sincronización y transferencia eficiente de archivos y directorios. Es ampliamente utilizada para crear copias de seguridad debido a su velocidad, flexibilidad y capacidad de sincronizar solo los cambios en los archivos.

A la hora de enviar los backups desde la máquina Linux a nuestro servidor utilizaremos scp (Secure Copy Protocol) y SSH. Scp es un comando de Linux que permite copiar archivos de manera segura entre sistemas remotos utilizando el protocolo SSH. SSH se utiliza para la autenticación y la transferencia de datos cifrada, lo que lo hace seguro para enviar backups a servidores remotos.

Con las herramientas mencionadas hasta ahora, podemos generar copias de seguridad y enviarlas a nuestro Windows Server. Sin embargo, este proceso tendría que realizarse manualmente. Para automatizarlo, crearemos un script en Linux que ejecute todas estas tareas de forma automática.

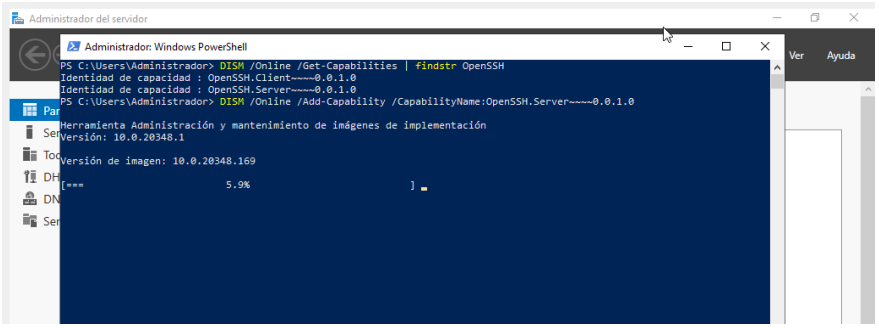
Además, utilizaremos Cron para programar la ejecución periódica del script, asegurando que el sistema funcione de manera completamente automatizada. Cron es un “daemon” de sistema en Linux que permite automatizar tareas programadas ejecutándolas en intervalos de tiempo específicos.

De este modo, el usuario no tendrá que preocuparse por realizar las copias de seguridad ni enviarlas manualmente.

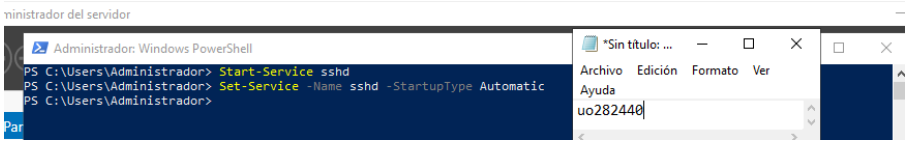
2.1 Instalar OpenSSH en Windows Server 2022

El primer paso para realizar los distintos ejercicios de este trabajo sería instalar openSSH en el servidor al que vamos a enviar las copias de seguridad, para que así podamos establecer una conexión segura a la hora de transferir nuestros Backups.

Instalamos OpenSSH:



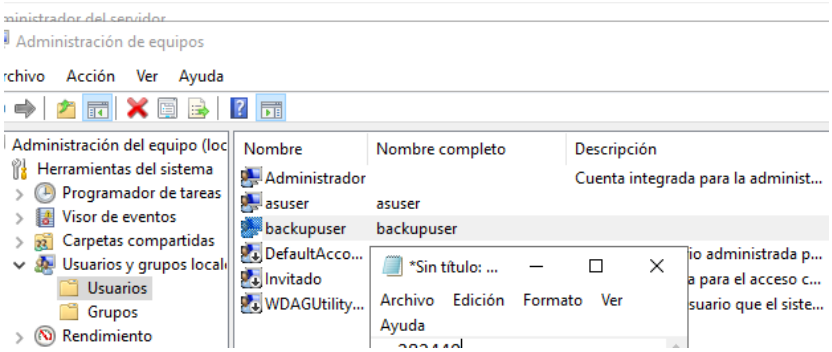
Iniciamos el servicio SSH y lo configuramos para que se inicie automáticamente:



Para que SSH funcione de forma correcta en nuestro Windows Server hay que añadir configuraciones adicionales, se debe modificar el Firewall para que permita las conexiones SSH, además se deberá crear un nuevo usuario para que reciba las copias de seguridad.

2.2 Creación de usuario en el servidor para recibir los Backups

A continuación, creamos el usuario “backupuser” en Windows Server 2022:



A este usuario se le deben otorgar los permisos correspondientes para que funcione correctamente SSH, y posteriormente se reinicia el servicio.

2.3 Configurar Linux para enviar las Copias de Seguridad

SSH utiliza un sistema de autenticación con claves en lugar de contraseñas para hacer las conexiones más seguras y automáticas. Para que un equipo Linux pueda conectarse sin necesidad de ingresar una contraseña cada vez, se usa un par de claves, la clave privada y la clave pública.

Esto nos permite conectarnos a nuestro Windows Server sin necesidad de contraseña y automatizar procesos.

Generamos las claves de SSH en nuestra máquina Linux:

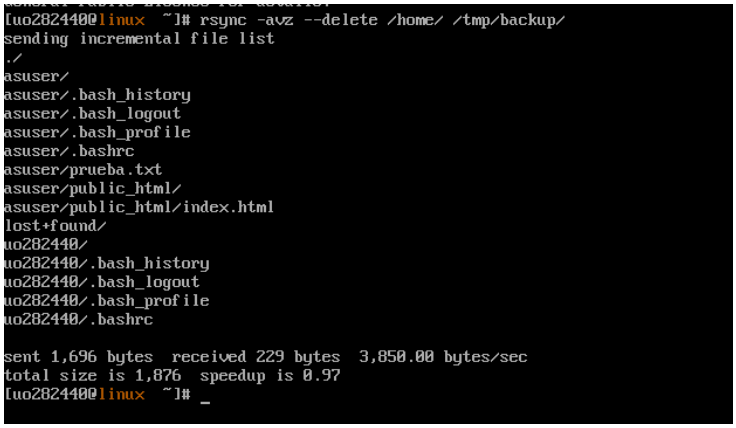
```
[uo282440@linux ~]# ssh-keygen -t rsa -b 4096 -C "backup-linux"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:3Ey0FTbYskHttWX1YAXHzOC23hR7O/RzQkOOIIzfXCw backup-linux
The key's randomart image is:
+---[RSA 4096]----+
|         .o+=.=B=|
|       o .+=o+.oB|
|      . o E=o.+=.|
|       o B.+.=..o|
|        S = . =oo|
|             o.++|
|              o++|
|               .+|
|                 |
+----[SHA256]-----+
[uo282440@linux ~]# _
```

Ahora copiamos la clave pública al servidor con el comando:
#ssh-copy-id backupuser@192.168.56.101

Y de esta manera ya tendríamos creado el sistema de comunicación segura entre nuestro Linux y nuestro servidor Windows.

2.4 Realizamos la Copia de Seguridad

A continuación, realizamos el Backup de forma manual con Rsync:



En este caso sincronizamos el contenido de /home en /tmp/backup

En este comando de Rsync, se utilizan parámetros como “–avz” o “--delete”, “-avz” es una combinación de opciones:

-a, Activa el modo archivo (archive), lo que significa que mantiene la estructura de directorios, permisos, propietarios, marcas de tiempo y enlaces simbólicos. Opción necesaria si queremos mantener la misma estructura en la copia de seguridad.

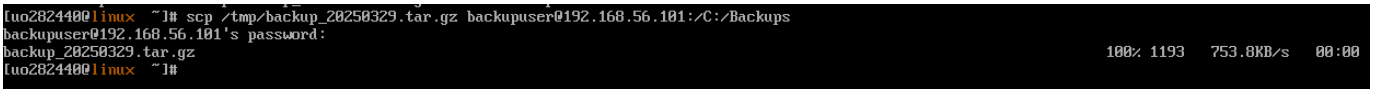
-v, “verbose” nos proporciona información detallada

-z, comprime los datos durante la transferencia

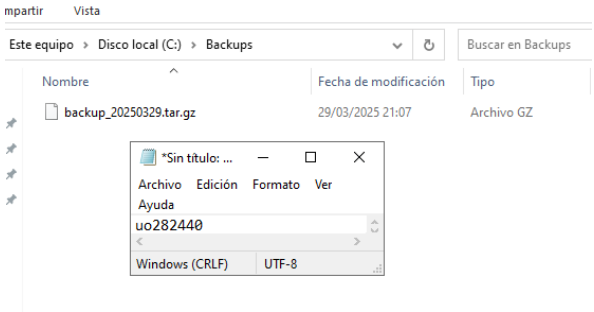
--delete, elimina información en el destino que ya no existe en el origen. Nos ayuda a generar una réplica exacta.

2.5 Enviamos la Copia de Seguridad al servidor

Después de modificar las propiedades de seguridad de la carpeta backups en Windows Server 2022 para que se pueda modificar su contenido, enviamos el backup manualmente:



Una vez lo hemos enviado, comprobamos que ha llegado correctamente:



Por último, eliminamos los archivos temporales de la copia de seguridad que hemos realizado antes y ya habríamos completado todo el proceso, ahora veremos cómo automatizarlo.

2.6 Automatizamos el sistema de Backups

Para automatizar todo el sistema de generación de copias de seguridad y enviarlas al servidor remoto, lo que podemos hacer es agrupar todos esos pasos y crear un Script que cuando se ejecute, realice todas esas esas tareas que hemos hecho ahora de forma manual pero automáticamente.

Para ello procedemos a crear el Script en Linux que agrupe todas las acciones anteriores:

```
 GNU nano 5.6.1                                          backup_to_W
#!/bin/bash


BACKUP_SRC="/home"
BACKUP_DEST_DIR="/tmp/backup"
WINDOWS_USER="backupuser"
WINDOWS_IP="192.168.56.101"
WINDOWS_PATH="/C:/Backups"


#creamos el directorio temporal
mkdir -p "$BACKUP_DEST_DIR"


#Sincronizamos con rsync
rsync -avz --delete "$BACKUP_SRC/" "$BACKUP_DEST_DIR/"


#comprimimos el backup
BACKUP_FILE="/tmp/backup_$(date +%Y%m%d).tar.gz"
tar -czvf "$BACKUP_FILE" -C "$BACKUP_DEST_DIR" .


#Lo enviamos a Windows server
scp "$BACKUP_FILE" "$WINDOWS_USER@$WINDOWS_IP:$WINDOWS_PATH"


#limpiamos archivos temporales
rm -rf "$BACKUP_DEST_DIR" "$BACKUP_FILE"


echo "Copia de Seguridad realizada y enviada a Windows Server 22"









                                                        [ 34 línea
^G Ayuda        ^O Guardar      ^W Buscar       ^K Cortar       ^T Ejecutar
```

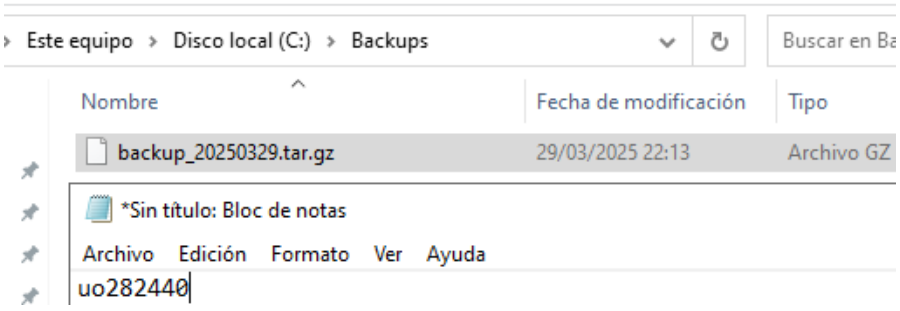
Ejecutamos el script:

```
[uo282440@linux  ~]#./backup_to_Windows_Server.sh
sending incremental file list
./
asuser/
asuser/.bash_history
asuser/.bash_logout
asuser/.bash_profile
asuser/.bashrc
asuser/prueba.txt
asuser/public_html/
asuser/public_html/index.html
lost+found/
uo282440/
uo282440/.bash_history
```

...

```
backup_20250329.tar.gz
Copia de Seguridad realizada y enviada a Windows Server 22
[uo282440@linux  ~]# _
```

Volvemos a comprobar que la nueva copia de seguridad ha llegado a Windows Server 2022:



Si queremos realizar copias de seguridad de forma periódica se podría hacer que este Script se ejecutase cada cierto tiempo, esto se consigue modificando el archivo crontab (#crontab -e) y especificando cuando queremos que se ejecute el Script.

Por ejemplo, si queremos hacer una copia de seguridad cada día a las dos de la mañana, tendríamos que añadir al final del archivo:
0 2 * * * /root/backup_to_Windows_Server.sh

Si quisiéramos tener una copia de seguridad todas las semanas cada domingo, añadiríamos la siguiente línea:
30 1 * * 0 /root/backup_to_Windows_Server.sh

También se podría, por ejemplo, ejecutar cada 6 horas si queremos tener muchas copias de seguridad:
0 */6 * * * / root/backup_to_Windows_Server.sh

Gracias a cron, podemos personalizar la frecuencia de los respaldos según las necesidades del sistema o del usuario, asegurando así que las copias de seguridad se realicen de manera automatizada y sin intervención manual.

3. Otras formas de realizar backups: Restic y Borg

En esta segunda parte del trabajo, vamos a analizar otras alternativas para hacer copias de seguridad, centrándonos en herramientas como Restic y Borg. Veremos sus ventajas y desventajas, compararemos sus características más importantes y evaluaremos cuál puede ser más útil según el caso.

Además, al final haremos un ejemplo práctico sobre cómo crear un sistema automatizado de backups con Restic, para entender mejor su funcionamiento y cómo se puede integrar en un entorno real.

El objetivo de este ejercicio es profundizar en opciones más avanzadas y ver qué herramientas son más utilizadas en la industria para gestionar backups de forma eficiente.

3.1 Restic vs Borg

Restic y Borg son herramientas potentes y muy utilizadas en la industria a la hora de realizar copias de seguridad. A continuación, veremos las distintas características que tienen estas herramientas y cuál puede adaptarse mejor a nuestras necesidades.

En cuanto a compatibilidad, Restic es superior a Borg, este último se usa principalmente en entornos Linux o en entornos Windows con WSL. Restic, en cambio, tiene compatibilidad con Linux, macOS y Windows, por lo que se podría ajustar mejor a nuestro caso.

Ambas herramientas emplean cifrado y deduplicación, una técnica que evita almacenar datos duplicados dentro de un sistema de copias de seguridad. En lugar de guardar múltiples copias idénticas de un archivo o fragmento de datos, el sistema solo almacena una versión única y crea referencias a ella cuando es necesario.

Se encuentran diferencias también cuando observamos su proceso de instalación, siendo el proceso de instalación de Restic más sencillo que el de Borg, el cual requiere más dependencias.

En cuanto a la velocidad, Restic puede ser más lento en sus backups iniciales y Borg es más rápido en términos generales, debido a su compresión y deduplicación optimizada.

A la hora de automatizar el proceso, es más sencillo realizarlo con Restic, ya que se puede automatizar con Cron y con Scripts, sin embargo, Borg requiere Borgmatic para realizar esta tarea de forma más sencilla.

Teniendo en cuenta lo anterior podríamos establecer que Restic es mejor opción para nuestro caso concreto, debido a su soporte nativo para Windows, más opciones de almacenamiento, instalación, automatización y uso más sencillo.

Por lo tanto, a continuación, se muestra cómo crear un sistema de copias de seguridad automatizado con Restic.

3.2 Realizamos el Backup utilizando Restic

Instalamos Restic desde el repositorio EPEL:

Restic no está disponible en los repositorios base predeterminados de esta distribución, por lo que utilizamos EPEL, un repositorio adicional que proporciona paquetes que no están oficialmente en RHEL/AlmaLinux.

3.2.2 Inicializamos el repositorio (destino del backup)

Restic necesita un lugar donde guardar los backups. Debido a su diseño y arquitectura se necita inicializar un repositorio, ya que no es una herramienta simple como Rsync, Restic nos crea un repositorio estructurado con características avanzadas como cifrado, dedeuplicación, snapshots, etc.

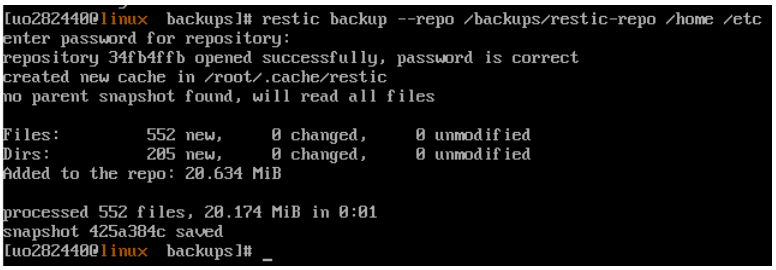
Al inicializarlo se crea la estructura del directorio base, se configura el cifrado automático y se habilita la deduplicación.

Inicializamos nuestro repositorio con Restic:

Como podemos ver Restic hace uso de las contraseñas, son un pilar fundamental en su diseño seguro, esto nos permite obtener un cifrado de extremo a extremo, por lo que sin la contraseña los datos no son legibles, aún si alguien accede al repositorio. También nos ayuda a prevenir accesos no autorizados o a verificar la integridad de los datos.

3.2.3 Realizamos el primer backup

En este caso haremos una copia de seguridad del contenido de los directorios /home y /etc y las guardaremos en nuestro repositorio.



Existen otras opciones útiles en restic para ser más precisos a la hora de realizar una copia de seguridad, cómo por ejemplo, el exclude (ignorar el directorio de descargas: --exclude="/home/usuario/descargas") o tag, para etiquetar el backup (--tag="mi-backup").

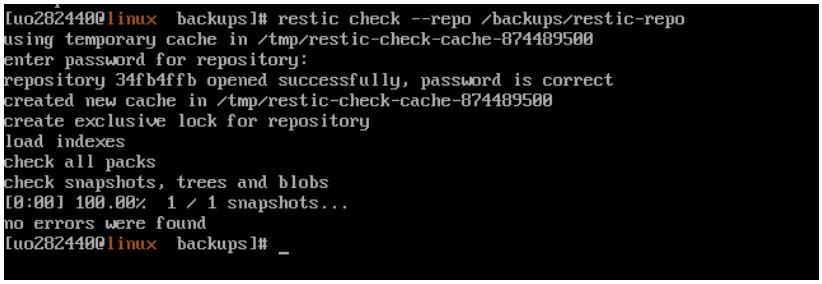
3.2.4 Verificamos la Copia de Seguridad creada

Comprobamos que la copia de seguridad se ha creado correctamente:

Verificar un backup tras haberlo creado es un paso crucial para garantizar su integridad, confiabilidad y recuperabilidad de los datos.

Esto nos permite detectar corrupción de forma temprana, problemas de almacenamiento, confirmar que el cifrado funciona, validar la deduplicación, etc.

Restic nos permite realizar todo esto mediante un único comando, “check”:



3.2.5 Automatizar:

Como hemos visto en la introducción de este segundo apartado, la automatización de copias de seguridad utilizando Restic se puede realizar de forma sencilla.

Al igual que en el primer ejemplo, donde implementamos un sistema de backups mediante Rsync, se podría lograr una automatización eficiente y programada periódicamente utilizando herramientas como Cron o Scripts personalizados.

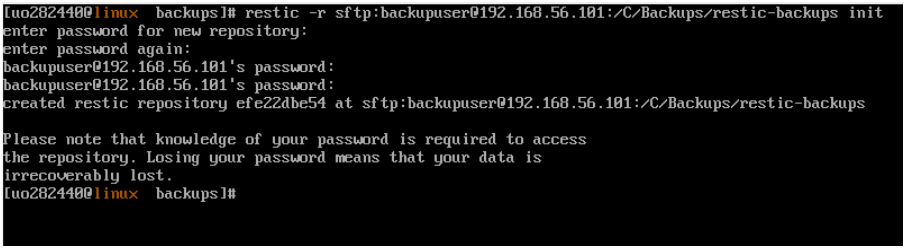
3.2.6 Envío a servidor remoto

Restic tiene soporte nativo para enviar backups a servidores remotos usando distintos protocolos. Puede enviar copias de seguridad a un servidor con protocolos como SFTP (SSH), WebDAV o almacenamiento en la nube**.**

Lo que hace es cifrar y comprimir los datos antes de enviarlos para asegurarse de que la información está protegida.

Para utilizar Restic de esta forma hay una serie de pasos que debemos seguir, al igual que en local, se configura un repositorio remoto donde se guardarán las copias de seguridad en el servidor. Restic analiza los archivos, los deduplica para evitar almacenar datos repetidos y los sube al servidor, de esta forma, cada vez que se ejecute un nuevo backup solo se envían los cambios en lugar de copiar todo de nuevo, esto hace que el proceso sea más rápido y eficiente.

Como hemos visto el primer paso es inicializar el repositorio remoto, se puede realizar esta acción sin ningún problema desde la propia máquina Linux.

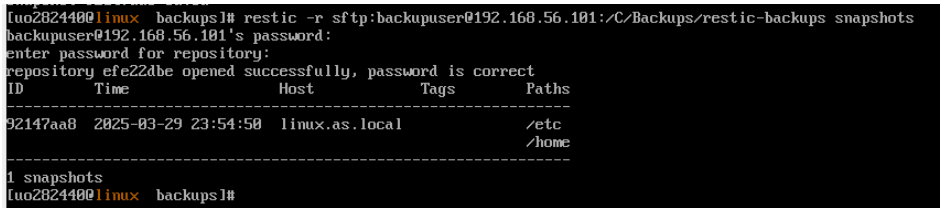


Una vez ya tenemos configurado el repositorio remoto que permita conexiones SFTP a través de SSH hacemos el backup de manera similar a la que vimos anteriormente, pero en este caso lo enviamos mediante SFTP y SSH.

Realizar este proceso mediante SFTP y SSH nos garantiza que los datos se transfieren de manera cifrada y autenticada.

Además, este método no requiere configuraciones adicionales de red y se puede programar fácilmente para que las copias de seguridad se realicen de manera recurrente sin intervención manual.

Una vez hemos realizado la copia de seguridad, verificamos que el backup se ha realizado correctamente y se encuentra en el servidor, esto se puede realizar también desde la propia máquina Linux con el comando de Restic “snapshots”:

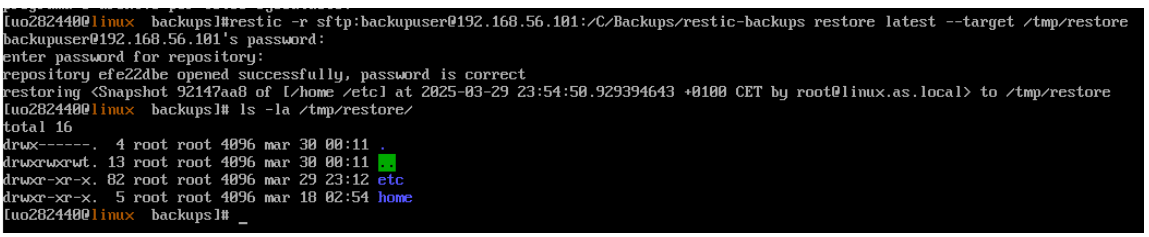


A continuación, veremos cómo restaurar una copia de seguridad de manera remota con Restic, para ello necesitaremos, como ya vimos anteriormente, conexión al servidor remoto, acceso al repositorio remoto donde se encuentran las copias de seguridad y seleccionar el snapshot que queramos.

Restic puede tener diferentes instantáneas(Snapshots) de nuestras copias de seguridad, con el comando visto anteriormente (restic snapshots), se pueden ver las distintas versiones disponibles de nuestros backups, incluyendo su fecha.

Una vez hemos elegido el snapshot podemos usar el comando “restore” de restic para restaurar nuestra copia de seguridad en nuestro sistema local.

Restic descarga y restaura los archivos del repositorio remoto al repositorio local que determinemos:



4. Conclusión

Gracias a la realización de este trabajo, he podido comprender mucho mejor cómo funciona el proceso de generar copias de seguridad de forma automática, así como la importancia de almacenar estos backups en un servidor remoto seguro. He explorado diferentes métodos para crear sistemas de copias de seguridad y he aprendido sobre diversas herramientas utilizadas en la industria, como Rsync, Restic y Borg.