**Relevamiento Sistema operativo CentOS.**

**VENTAJAS y BENEFICIOS:**

A continuación se presentarán varios aspectos positivos los cuales fueron decisivos a la hora de la elección de dicho sistema operativo para su puesta en marcha en un entorno infraestructural de servidores empresariales.

Costo: CentOS es una distribución de código abierto y es gratuito para descargar, instalar y utilizar. Esto hace que CentOS sea una opción atractiva para aquellos que buscan una alternativa de bajo costo a otros sistemas operativos comerciales.

Flexibilidad: CentOS es altamente flexible y se puede adaptar a diferentes necesidades y casos de uso. Se puede utilizar tanto en entornos de servidor físicos como virtuales, y ofrece opciones de personalización para adaptarse a las preferencias y requisitos específicos de cada usuario.

Comunidad de soporte: CentOS cuenta con una gran comunidad de usuarios y desarrolladores que brindan soporte y comparten conocimientos a través de foros, wikis y otros canales de comunicación. Esto proporciona a los usuarios acceso a recursos valiosos y ayuda en caso de problemas o preguntas.

Estabilidad: CentOS está diseñado para ser estable y confiable. Se basa en el código fuente de RHEL, que es conocido por su robustez y rendimiento en entornos de producción.

Soporte a Largo Plazo: CentOS sigue un ciclo de vida de soporte a largo plazo, lo que significa que cada versión principal se mantiene y recibe actualizaciones durante un período prolongado. Esto es especialmente beneficioso para los entornos empresariales, ya que brinda estabilidad y evita la necesidad de actualizaciones frecuentes que podrían interrumpir el funcionamiento de los servidores.

**APARTADO SOPORTE TECNICO:**

En está sección se abarcara toda la documentación con el que contara el equipo instalador, con los pasos y/o requisitos de dicho entorno de servidor.

Monitoreo y Supervisión:

Se dará como implementado herramientas de monitoreo para la supervisión del rendimiento de dicho servidor incluyendo así datos no prescindibles como: Estado de uso de CPU, memoria, unidad de almacenamiento y uso de red ethernet.

Se configuraran alertas para dar como notificado al equipo de soporte técnico ante cualquier tipo de problema inesperado tales como: alta carga en CPU, escasez de memoria, espacio en unidad de almacenamiento insuficiente o problemas de conectividad con la red actualmente en uso.

Actualizaciones Y Parches:

Mantener el servidor actualizado con las últimas actualizaciones y parches de seguridad de CentOS. Esto garantizará que el servidor esté protegido contra vulnerabilidades conocidas y que se beneficien de las mejoras y correcciones de errores.

Establecer un horario regular para aplicar las actualizaciones y parches, teniendo en cuenta los momentos de menor actividad del servidor para minimizar la interrupción del servicio.

Seguridad:

Configurar un firewall para restringir el acceso no autorizado al servidor y asegurarse de que solo los puertos necesarios estén abiertos.

Implementar medidas de seguridad adicionales, como SELinux y herramientas de detección y prevención de intrusiones (IDS/IPS), para proteger el servidor contra amenazas y ataques externos.

Realizar auditorías de seguridad periódicas para identificar posibles brechas de seguridad y aplicar las correcciones correspondientes.

Supervisión de Rendimiento:

Realizar análisis periódicos del rendimiento del servidor y optimizar la configuración según sea necesario para garantizar un rendimiento óptimo.

Identificar cuellos de botella de rendimiento y aplicar soluciones, como ajustes de configuración o aasactualización de hardware, para mejorar el rendimiento del servidor

**Relevamiento Sistema operativo Windows 10**

En el siguiente apartado documentativo se presentaran las razones por el cual el equipo de soporte técnico tomó como decisión definitiva utilizar el sistema operativo Windows 10 dentro de las terminales a nivel usuario/cliente.

Interfaz de usuario moderna y de fácil uso a nivel usuario.

Windows 10 cuenta con una interfaz de usuario actualizada y eficiente que resulta sencilla de utilizar y personalizar. Los usuarios tienen la opción de personalizar su escritorio, modificar la apariencia del menú Inicio y acceder rápidamente a la configuración de manera simple.

Actualizaciones recurrentes al sistema operativo.

Microsoft garantiza actualizaciones periódicas que incluyen mejoras de funcionalidades y medidas de seguridad para mantener Windows 10 actualizado y protegido contra posibles riesgos. Los usuarios pueden estar tranquilos sabiendo que su sistema operativo se mantiene al día y está resguardado frente a las más recientes amenazas de seguridad.

Mejoras de rendimiento.

Cuenta con mejoras significativas en el rendimiento en comparación con versiones anteriores de Windows El sistema operativo es más rápido y eficiente, lo que permite a los usuarios realizar tareas más rápido y sin retrasos o interrupciones.

**APARTADO SOPORTE TECNICO WINDOWS 10:**

En el siguiente apartado se darán a conocer la pautas y/o practicas correctas para el mantenimiento prolongado de terminales operativas enfocadas a usuarios las cuales cuenten con el sistema operativo Windows 10 funcionando dentro de ellas.

Actualizar el sistema operativo:

Es importante mantener Windows 10 actualizado para recibir las últimas correcciones de seguridad y mejoras. Verifique si hay actualizaciones pendientes a través de "Configuración" -> "Actualización y seguridad" -> "Windows Update" y aplique las actualizaciones necesarias.

Ejecutar el solucionador de problemas:

Windows 10 incluye una función de solución de problemas integrada que puede ayudar a identificar y solucionar problemas comunes. Puedes acceder a ella a través de "Configuración" -> "Actualización y seguridad" -> "Solucionar problemas". Allí encontrarás diferentes opciones, como solucionadores de red, audio, impresión, etc.

Realizar un escaneo de malware:

 Si sospechas que el sistema puede estar infectado con malware, es recomendable realizar un escaneo completo utilizando un programa antivirus confiable. Asegúrate de que tu software antivirus esté actualizado y realiza un análisis exhaustivo del sistema.

Restauración del sistema:

Si los problemas en la terminal persisten y sospechas que pueden estar relacionados a cambios recientes, puede ser valido el uso de la función de Restaurar sistema. Esta opción permite al usuario regresar el sistema a un punto anterior en el tiempo en el que funcionaba correctamente. La manera de acceder a ella es a través de: "Configuración" -> "Actualización y seguridad " -> "Recuperación". (En el caso de contar con información la cual no se quiere ser perdida se recomienda el hacer una copia de seguridad pertinente).

**MANUAL DE INSTALACION SERVIDOR LINUX**

**MODULO INTRODUCTORIO:**

El siguiente manual tiene como objetivo principal el proporcionar una guía de instrucciones detalladas para la correcta instalación básica y puesta en marcha de un servidor Linux. En el mismo se utilizara como ejemplo practico el uso del sistema operativo basado en Unix llamado CentOS, el cual es ampliamente utilizado en entornos empresariales.

Requisitos previos.

Antes de dar por iniciada la instalacion se deberan de tener en cuenta los siguientes elementos:

1. Imagen de instalacion: Descargar la imagen de instalacion del sistema operativo en cuestion desde el sitio web oficial del proveedor.
2. Hardware compatible: Debe de ser verificado que el hardware a utilizar cumpla con los requisitos minimos del sistema especificado.
3. Soporte Tecnico: Asegurarse de contar con el acceso al soporte tecnico brindado por el proveedor del sistema operativo (en esté caso puede ser tomado como valida el uso de foros y/o comunidades formados por usuarios que utilizan dicho sistema).

**Procedimiento de Instalación:**

Preparación previa:

1. Verificar la integridad de la imagen descargada proporcionada por el proveedor.
2. Crear una copia de seguridad de cualquier tipo de dato de importancia (este paso puede ser omitido si se quiere realizar una instalacion limpia).
3. Realizar la conexión de perifericos necesarios adecuadamente.

BIOS:

1. Asegurarse que la secuencia de arranque inicial se encuentre configurada correctamente para que nuestro servidor arranque desde el medio de instalacion deseado (DVD/CD o Unidad booteable USB).
2. Realizar cualquier otra configuracion necesaria dentro de la BIOS según las recomendaciones del proveedor.

Instalacion:

1. Inserte el medio de instalacion escogido anteriormente y proceda a reiniciar el servidor.
2. Seleccióne la opcion de arranque desde el medio de instalacion en el menu de inicio.
3. Un vez iniciado el proceso, seleccione el idioma y otras configuraciones deseadas según su preferencia.

Particionado y configuracion de almacenamiento:

1. a. En la pantalla de particionado, seleccione el esquema de partición que mejor se adapte a sus necesidades o utilice la configuración predeterminada recomendada.
2. Realice las particiones para el sistema de achivos raiz (/) y cualquier otra particion deseada.
3. Aplique los cambios y confirme el particionado.

Configuración de red:

1. Configure la configuración de red según las necesidades de su entorno empresarial (dirección IP estática, DHCP, DNS, etc.).
2. Asigne un nombre de host adecuado para el servidor y configúrelo correctamente.

Apartado de seguridad y finalizacion:

1. Establezca una contraseña segura para la cuenta de administrador del sistema y cee cualquiera otra cuenta adicional de su preferencia.
2. Configure el firewall y otras medidas de seguridad según su criterio.
3. Finalize la instalacion aceptando todos los cambios anteriormente mencioandos en puntos posteriores.

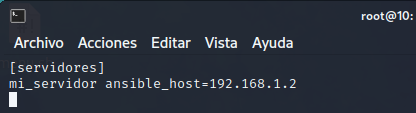
**INSTALACION ANSIBLE EN TERMINAL**

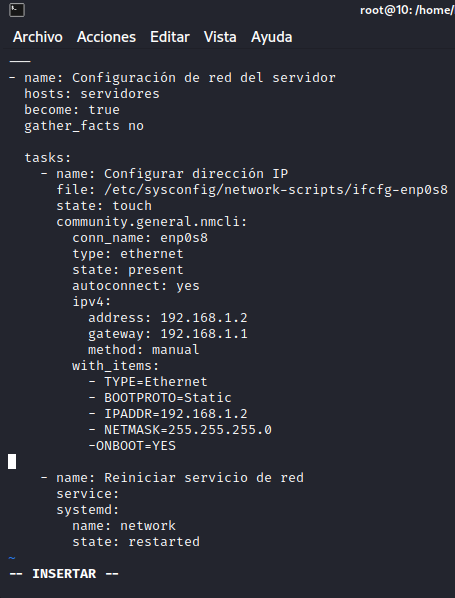
Guia basica para la correcta instalacion y puesta en marcha de ansible en una terminal con CentOS 7 instalado por medio de comandos pasado en consola.

1. Abrir la terminal de comandos.
2. Hacer el correcto login como usuario root utilizando el comando: sudo su.
3. En consola ejecutamos el comando: yum install epel-release para acceder a los paquetes necesarios para la instalación de ansible
4. En consola utilizar el comando: yum install ansible para iniciar su instalación
5. Una vez instalado utilizaremos el comando: ansible - -version el cual nos mostrara en pantalla la versión de ansible que fue instalada además de si ansible fue instalado correctamente.

**Guía de configuración red utilizando ansible:**

1. Crea un archivo de inventario de Ansible (por ejemplo, mi\_inventario.ini) y agrega la dirección IP del servidor al que deseas aplicar la configuración:

****

1. Creamos un archivo utilizando el gestor VI de tipo playbook de ansible como por ejemplo: (vi configuracion\_red\_servidor.yaml) el mismo contendrá todos los parámetros necesarios para la configuración de nuestra red dirigida la IP asignada tal como se muestra en la siguiente imagen.  ****

1. Utilizando el comando ansible-playbook -i mi\_inventario.ini configuracion\_red\_servidor.yaml le diremos al playbook ejecutarse en el servidor especifico el que se encuentra dentro de el archivo mi inventario creado en pasos anteriores.****

**Configuracion SSH.**

A continuación se proporciona una guía detallada sobre el uso de un playbook de Ansible diseñado para configurar un servidor SSH, instalar el cliente de MySQL, ajustar la conexión a la base de datos y agregar una clave SSH para un usuario específico en un servidor CentOS 7. Este playbook es una herramienta poderosa para la administración de servidores, ya que automatiza tareas comunes de configuración y permite una gestión eficiente y coherente de servidores.

**Funcionalidades del Playbook.**

**Configuración de Puerto SSH:**

Permite establecer un puerto personalizado para la conexión SSH. Esta característica es útil cuando se necesita cambiar el puerto predeterminado para mejorar la seguridad del servidor.

**Instalación del Cliente de MySQL:**

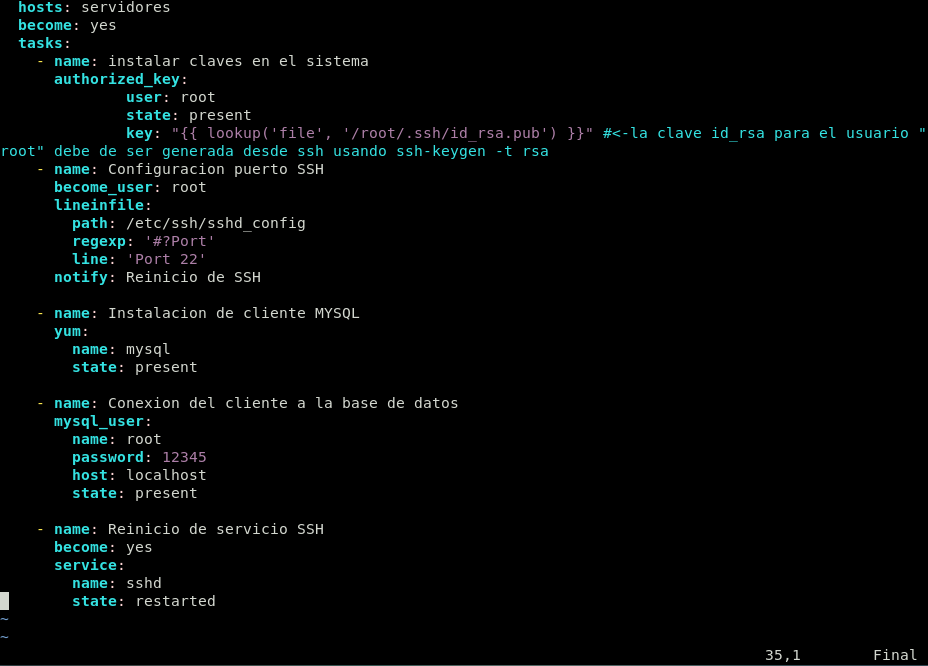
Nuestro playbook de Ansible simplifica el proceso de instalación del cliente de MySQL en el servidor Centos7. Esta instalación es esencial para permitir la interacción efectiva con bases de datos MySQL desde el servidor. Haciendo innecesario realizar este proceso manualmente, lo que ahorra tiempo y garantiza que el cliente de MySQL esté disponible y listo para su uso.

**Gestión de Claves SSH:**

Proporciona la capacidad de agregar una clave SSH a un usuario específico. Esto simplifica el proceso de autenticación y permite el acceso seguro al servidor.

**Ajustes de conexión a la Base de Datos:**

Permite configurar y ajustar la conexión a una base de datos MySQL. Esto incluye la creación de usuarios, contraseñas y permisos necesarios para acceder a la base de datos.

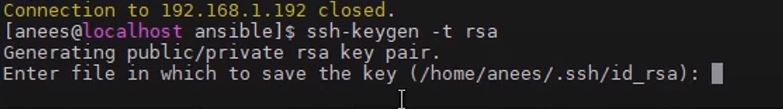


**Generacion de clave o “Key”**

Si queremos autorizar al usuario que escogimos (en este caso el usuario llamado “root”) al momento de realizar nuestro playbook al servicio ssh es necesario generar una clave para dicho usuario.

Esto quiere decir que debemos de hacer login dentro de ssh con un usuario del tipo administrador y proceder a escribir el siguiente comando:

**ssh-keygen -t rsa:** Este comando genera la clave o “key” la que luego se almacenara de un archivo el cual nosotros escogamos la ruta. En el siguiente ejemplo se muestra la ruta /home/anees/.ssh/id\_rsa dicha ruta deberá de ser colocada dentro de la línea **key:** de nuestro playbook. De esa manera autorizamos al usuario escogido anterior mente en la línea **user:** a acceder al servicio ssh.



**Medios de respaldo de datos y disponibilidad.**

En la gestión de servidores CentOS7, es esencial contar con una estrategia sólida para respaldar datos a largo plazo y garantizar una alta disponibilidad, se explorarán las estrategias fundamentales en esta área, centrándonos en cinco aspectos clave.

**Almacenamiento Redundante:**

La implementación de un sistema de almacenamiento RAID es fundamental. RAID 1, conocido como espejo, ofrece redundancia al duplicar los datos en dos discos duros o más. Por otro lado, RAID 6 utiliza paridad distribuida para brindar tolerancia a fallos incluso en la falla de dos discos. Estas configuraciones protegen contra pérdidas de datos y garantizan la disponibilidad continua.

**Copias de Seguridad Regulares:**

La realización de copias de seguridad regulares es esencial para la protección de datos. Herramientas como rsync y tar son útiles para este propósito. Además, almacenar copias de seguridad en ubicaciones geográficamente separadas reduce el riesgo de pérdida de datos debido a desastres naturales.

**Virtualización y Migración:**

La virtualización de servidores permite la fácil migración de sistemas y datos en caso de fallo del servidor físico. Esto proporciona una alta disponibilidad al minimizar el tiempo de inactividad en situaciones de emergencia. La virtualización puede lograrse mediante soluciones como VMware o VirtualBox.

**Encriptación de Datos:**

La encriptación de datos es crucial tanto en reposo como en tránsito. Garantizar que los datos estén protegidos contra accesos no autorizados es esencial para la seguridad de los datos. El uso de protocolos seguros como HTTPS para la transmisión de datos y la implementación de sistemas de archivos encriptados son prácticas recomendadas.

**Actualizaciones y Parches:**

El mantenimiento constante del sistema es fundamental para mitigar vulnerabilidades y garantizar un rendimiento óptimo. Se deben aplicar regularmente actualizaciones de seguridad y parches para mantener el servidor seguro y en funcionamiento sin problemas.

**Conclusión:**

La combinación de almacenamiento redundante, copias de seguridad regulares, virtualización y migración, encriptación de datos y mantenimiento del sistema en servidores brinda una estrategia para respaldar datos a largo plazo y garantizar una alta disponibilidad. Estas prácticas son esenciales para proteger la accesibilidad de los datos. La implementación adecuada de estas estrategias proporciona la confianza necesaria para enfrentar los desafíos que puedan surgir en la gestión de servidores.

**Rutinas de Backups.**

A continuación se procederá de forma detallada la puesta en uso del script el cual llamamos “Backups\_rutina.sh” de modo que dicho script sea ejecutado automáticamente por nuestro sistemas operativo centos7 sin la necesidad de la presencia de un operador en el momento, para esto se deberá de contar con el el uso del programa “cron” el cual nos proporcionará dicha automatización.

Deberemos de crear archivo utilizando el comando touch nombrando dicho archivo con su extensión .sh, Ingresamos al archivo utilizando el gestor de textos vi. Una vez dentro comenzamos a incluir los siguientes comandos:

**#!/bin/bash:** Nos permitirá que dicho archivo sea interpretado como un script por bash.

**Hora=$(date + ‘%H:%M%:S’):** Nos permite crear una variable llamada “hora” la cual nos dira la hora exacta.

**“$Hora”:** Variable a utilizar

Echo -e: Comando en la terminal de centos7 que se utiliza para imprimir texto en pantalla.

**Playbooks\_dir=”$HOME/etc/ansible/hosts”:**

Playbooks\_dir: es una variable que almacena una ruta de directorio.

**$HOME** es una variable de entorno que generalmente representa el directorio de inicio del usuario.

La ruta completa etc/ansible/hosts se agrega al directorio de inicio usando $HOME. (**IMPORTANTE:** utilizaremos la ruta etc/ansible/hosts ya que es la ruta creada por defectos en ansible en la cual se almacenan dichos playbooks)

**Backup\_dir=”$HOME/ruta\_del\_backup/backups/Carpeta\_de\_playbooks”:**

Backup\_dir es otra variable que almacena una ruta de directorio.

**Mkdir -p “$backup\_dir”:** En caso de no existir todavía un directorio el cual alamcene nuestros playbooks, este comando creara uno.

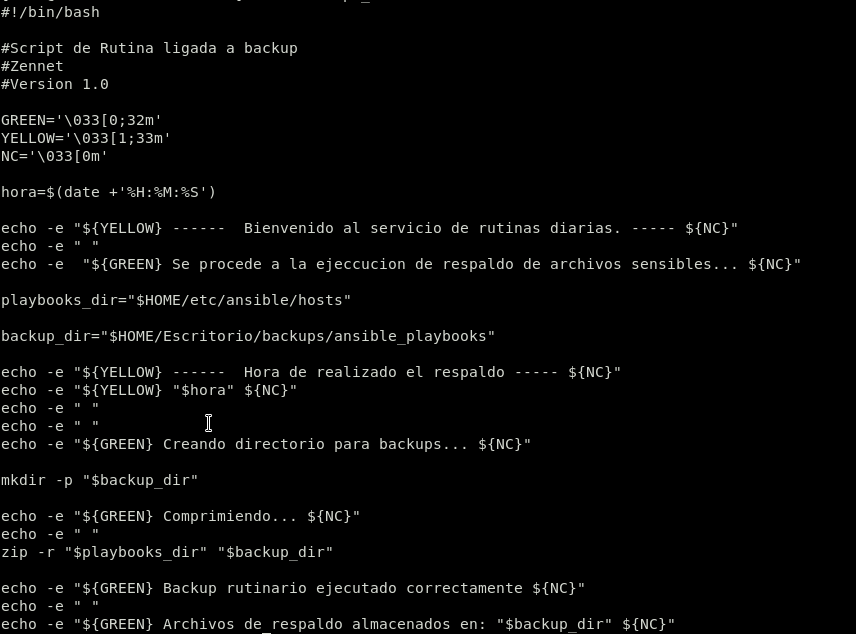
**Comando Adicional:**

**Zip -r “$playbooks\_dir” “$backup\_dir”:** Permite comprimir nuestro backup y lo envia a el directorio el cual queremos se almacene usando las variables anteriormente creadas.

Zip: comando que se utiliza para crear archivos ZIP. (Es importante asegurarse de que el comando zip esté instalado en tu sistema antes de usarlo).

-r: Esta opción indica a zip que debe comprimir recursivamente todos los archivos y subdirectorios dentro del directorio que deseas comprimir.

**Representación grafica del contenido del script.**



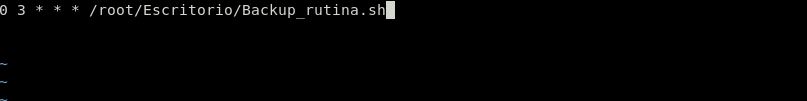


**Creación de rutina utilizando cron:**

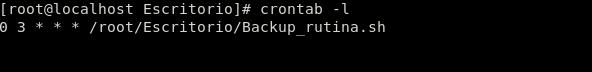
1. Ejecutamos el programa cron utilizando el comando: crontab -e



1. automáticamente nos dirigirá a un archivo con el editor de vi tal como se muestra en la siguiente imagen:



1. Nuestro siguiente paso para realizar la rutina será el seteo de el horario el cual querramos definir para el inicio automático de nuestro script. Para este ejemplo nuesto script se ejecutará diariamente a la hora: 03:00am. (Para realizar el calculo necesario puede consultarse de manera gratuita por medio del siguiente link: [AQUI](https://crontab.guru/#*_*_*_*_*)) Luego se procede a definir la ruta de nuestro archivo en nuestro caso, /root/Escritorio/Backup\_rutina.sh.
2. Al finalizar nuestra tareas usaremos el comando WQ! Que nos permitirá guardar y salir de crontab, una vez hecho nuestra tarea quedará programada con éxito.
3. En caso de querer verificar si nuestra/s rutinas se encuentran operativas utilizaremos el comando: crontab -L el cual desplegara la lista de tareas en ejecución.



**Backup de BDD.**

A continuación se muestra el playbook creado en el gestor de textos vi que nos permitirá el realizar exitosamente el backup o respaldo de una base de datos, para este ejemplo tomamos la base de datos llamada “MariaDB” utilizando el siguiente comando:

**mysqldump:** Este es el comando que se utiliza para ejecutar la utilidad de copia de seguridad de MySQL.

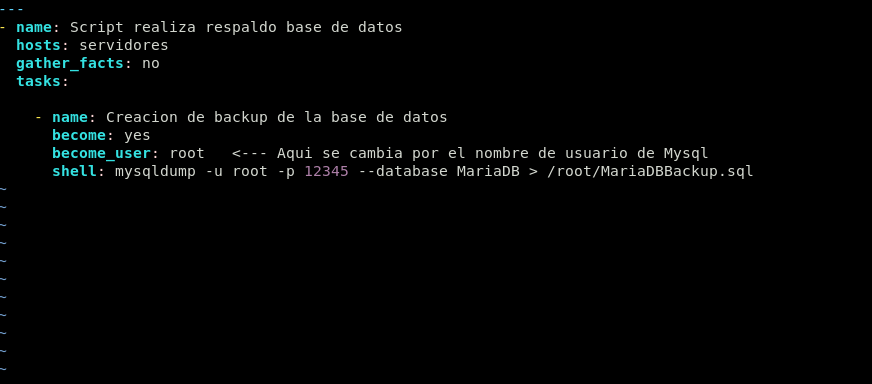
**-u root**: Esto especifica el usuario de MySQL que se utilizará para realizar la copia de seguridad. En este caso, se está utilizando el usuario “root”.

-**p:** Esta opción indica a MySQL que debe solicitar una contraseña. Después de ingresar el comando, el sistema te pedirá que ingreses la contraseña la cual en este caso fue “12345”

**--database MariaDB**: Esta parte del comando especifica la base de datos que se va a respaldar. En este caso, se está respaldando una base de datos llamada “MariaDB”, (Debes reemplazar “MariaDB” con el nombre de la base de datos que deseas respaldar).

>: Operador lógico que redirige la salida del comando a un archivo.

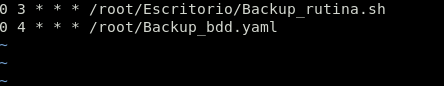
**/root/MariaDBBackup.sql**: Esta es la ruta y el nombre del archivo de respaldo donde se guardarán los datos de la base de datos.



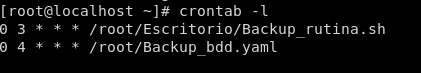
**Creacion de rutina:** Al igual que en nuestro caso anterior se creará una tarea utilizando cron la cual permita ejectuar dicho playbook de forma automática ya sea en una fecha, hora o tiempo en especifico.

Al momento de crear nuestra nueva tarea de rutina no es necesario crear un nuevo archivo crontab ya que cron posee una arquitectura la cual puede ejecutar varias tareas sean del formato o tipo que sean.

Con esto en mente procedemos a utilizar el comando: crontab -e y redactar la automatización de nuestra tarea en una nueva línea tal como se muestra en la imagen:



Para este ejemplo se dejo un margen de unos 60 minutos entre backups ya que se busca el ahorro de recursos de nuestro servidor. En caso de ser necesario podremos verificar el estado de nuestras tareas utilizand crontab -l.



**Script centro de computos.**

**Introducción.**

El script en cuestión desempeña un papel fundamental al proporcionar información detallada sobre nuestro sistema. Esto tiene diversas aplicaciones útiles tales como:

**Monitoreo del Sistema:** Permite supervisar el estado general del sistema. Los administradores pueden usarlo para mantener un ojo en la salud del servidor.

**Resolución de Problemas:** En situaciones de diagnóstico y resolución de problemas, el script brinda detalles valiosos sobre la configuración del sistema y el uso de recursos, lo que ayuda a identificar posibles problemas.

**Verificación de Recursos:** Es útil para verificar cómo se están utilizando recursos críticos como la CPU, la memoria y el espacio en disco, lo que es esencial para garantizar que el servidor funcione de manera eficiente.

**Documentación y Registro:** Facilita la documentación y el registro de la configuración y el estado del sistema en un momento dado. Esto es beneficioso para el mantenimiento y la auditoría del sistema.

En resumen, este script es una herramienta versátil que proporciona información rápida y relevante, Esto es fundamental para la monitorización, el diagnóstico de problemas y la gestión efectiva de recursos.

**Uso de comandos dentro del Script.**

**echo -e “Hostname: $(hostname)”**

Muestra el nombre del host del sistema, seguido por el nombre del host obtenido mediante el comando hostname.

**echo -e "Sistema Operativo: $(cat /etc/os-release | grep “PRETTY\_NAME” | cut -d”” -f )”**

Muestra el sistema operativo, seguido por el nombre del sistema operativo obtenido del archivo /etc/os-release.

**echo -e “Kernel: $(uname -r)”**

Muestra la versión del kernel, obtenida mediante el comando uname -r.

**echo -e “Memoria Total: $(free -m | awk 'NR==2 {print $2 “ MB”}')”**

Muestra la cantidad total de memoria, obtenida del comando free -m.

**\*echo -e “CPU: $(lscpu | grep “Model name” | cut -d':' -f 2 | sed 's/^ //')”**

Muestra información sobre la CPU, incluyendo el modelo de la CPU. La información se obtiene del comando lscpu.

echo -e "Memoria Usada: **$(free -m | grep “Mem:”** | awk '{print $3 “MB”}') / Memoria Total: **$(free -m | grep “Mem:”** **| awk '{print $2 “ MB”}')”**

Muestra la cantidad de memoria usada y la memoria total. Los valores se obtienen del comando free.

**Who:**

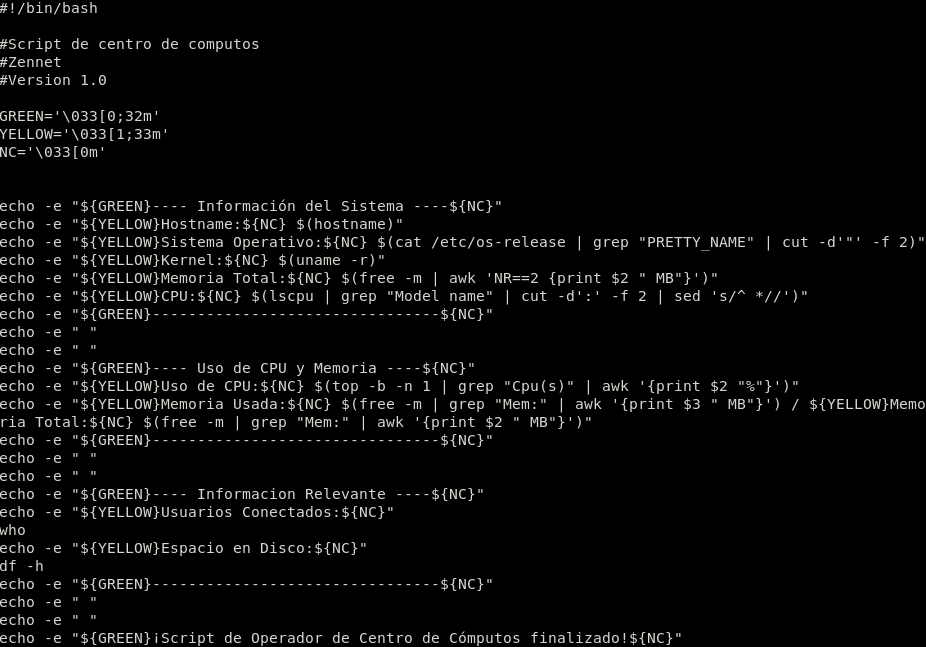
Ejecuta el comando who para mostrar información detallada sobre los usuarios conectados.

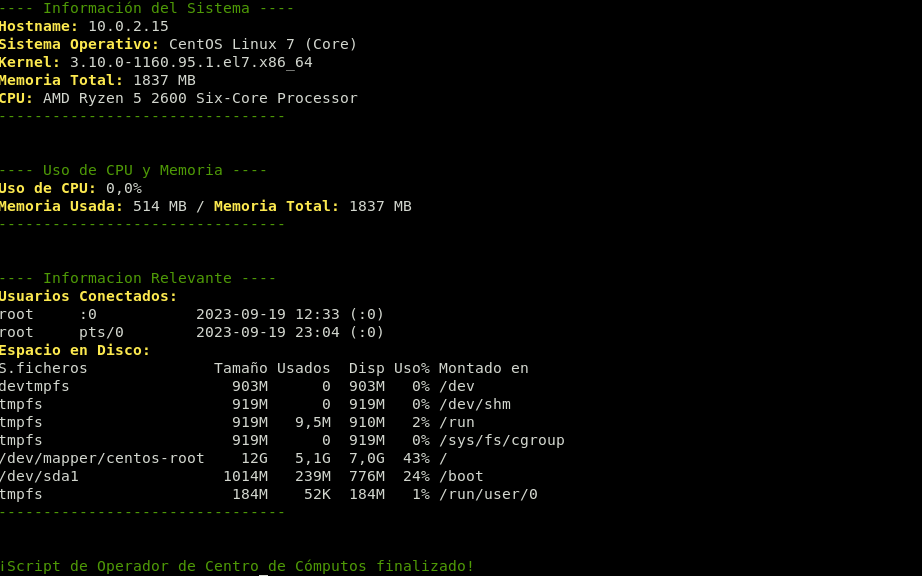
**df -h:**

El comando df -h muestra una lista de sistemas de archivos montados y su uso de espacio en disco.

**Perspectiva grafica.**

Imágenes del script en cuestión puesto en función obteniendo así la información del sistema.





**Shell script modular.**

**Introduccion a Shell script.**

Un shell script es un archivo de texto que contiene una secuencia de comandos que se ejecutan en una línea de comandos de un sistema operativo Unix o Unix-like, como Linux. La modularidad en un shell script se refiere a la práctica de dividir el script en módulos o funciones independientes, lo que facilita la organización y el mantenimiento del código.

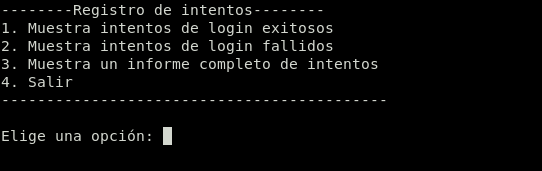
**Declaración de intérprete:** El script comienza con la especificación del intérprete que se utilizará para ejecutar el código, comúnmente **#!/bin/bash** para indicar que se utilizará el shell Bash.

**Módulos o funciones:** El script se divide en módulos o funciones independientes que realizan tareas específicas. Cada función se define con el formato.

**Formacion del Shell script:** El siguiente script modular contendrá varios scripts de forma fragmentada/modular los cuales realizaran la tarea de recolectar los intentos de login a nuestra terminal o sistema. Dicho script se separa en cuatro modulos, dos son destinados a los intentos de login, uno destinado a nuestro menú y por ultimo tendremos nuestro script principal que se encarga de llamar a cada modulo individual.

**Menu.**

Opciones del Menú: El menú presenta cuatro opciones numeradas.

Solicitud de Elección: Después de mostrar las opciones, el script utiliza read -p para solicitar al usuario que elija una opción ingresando un número. La variable opcion almacenará la elección del usuario. 

**Principal.**

**Fuentes (Sources):**

El script utiliza las instrucciones source para incluir otros archivos de script en su ejecución.

source menu.sh incluye un archivo llamado menu.sh, que contiene funciones relacionadas con la interfaz de usuario.

source login\_exitoso.sh incluye un archivo llamado login\_exitoso.sh, que contiene funciones para analizar registros de inicio de sesión exitosos.

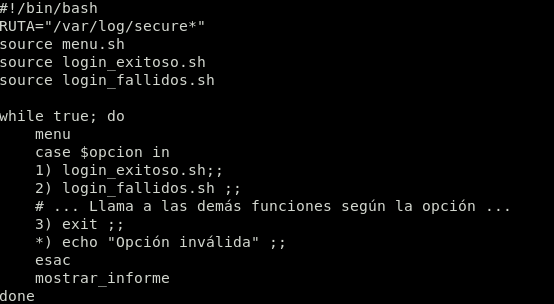
source login\_fallidos.sh incluye un archivo llamado login\_fallidos.sh, que contiene funciones para analizar registros de inicio de sesión fallidos.

Bucle Principal:

El script contiene un bucle while que se ejecuta indefinidamente (while true) para mantener el programa en funcionamiento hasta que el usuario decida salir.En cada iteración del bucle, se llama a la función menu para mostrar un menú interactivo al usuario y se le pide que elija una opción.Un case se utiliza para manejar la opción seleccionada por el usuario.

Ejecución de Funciones:

Dependiendo de la opción elegida por el usuario, se llaman a funciones específicas. Por ejemplo, si el usuario elige la opción 1, se llama a la función login\_exitoso para analizar registros de inicio de sesión exitosos. Lo mismo ocurre para la opción 2 y la función login\_fallidos.



**Entradas exitosas.**

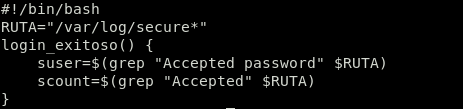
La función login\_exitoso se encarga de realizar el siguiente proceso.

Variables Internas:

Dentro de la función, se definen dos variables locales:

suser: Esta variable se utiliza para almacenar las líneas de registro que contienen la cadena "Accepted password". Estas líneas indican que un usuario ha tenido un inicio de sesión exitoso utilizando una contraseña.

scount: Esta variable se utiliza para almacenar todas las líneas de registro que contienen la cadena "Accepted". Esto incluye tanto los inicios de sesión exitosos con contraseña como otros métodos de autenticación exitosos.

Comandos grep:La función utiliza comandos grep para buscar cadenas específicas en el archivo de registro definido por la variable $RUTA. grep "Accepted password" $RUTA busca las líneas de registro que contienen la cadena "Accepted password" y las almacena en la variable suser. grep "Accepted" $RUTA busca todas las líneas de registro que contienen la cadena "Accepted" y las almacena en la variable scount.

**Entradas fallidas.**

La función login\_fallidos se encarga de realizar el siguiente proceso:

Variables Internas:

Dentro de la función, se definen tres variables locales.

tuser: Esta variable se utiliza para almacenar las líneas de registro que contienen las cadenas "Accepted" o "Failed". Estas líneas indican intentos de inicio de sesión exitosos o fallidos.

fuser: Esta variable se utiliza para almacenar las líneas de registro que contienen la cadena "Failed password". Estas líneas indican intentos de inicio de sesión fallidos debido a contraseñas incorrectas.

fcount: Esta variable se utiliza para almacenar todas las líneas de registro que contienen la cadena "Failed". Esto incluye tanto los intentos de inicio de sesión fallidos con contraseñas incorrectas como otros tipos de fallas.

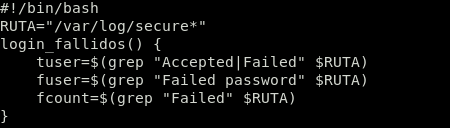
Comandos grep:

La función utiliza comandos grep para buscar cadenas específicas en el archivo de registro definido por la variable $RUTA.

grep "Accepted|Failed" $RUTA busca las líneas de registro que contienen las cadenas "Accepted" o "Failed" y las almacena en la variable tuser.

grep "Failed password" $RUTA busca las líneas de registro que contienen la cadena "Failed password" y las almacena en la variable fuser.

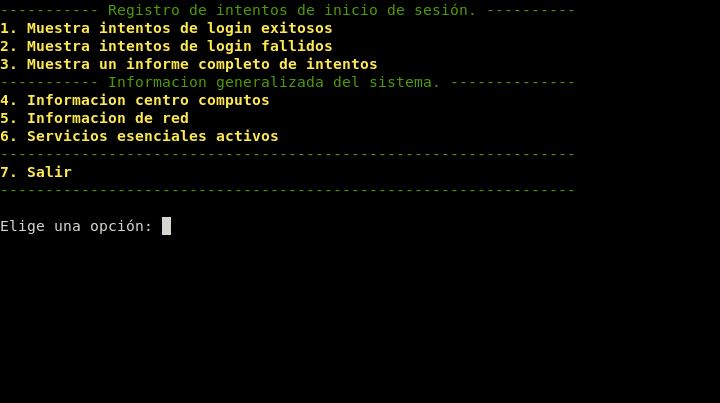
grep "Failed" $RUTA busca todas las líneas de registro que contienen la cadena "Failed" y las almacena en la variable fcount.



**INTEGRACION DE SCRIPTS FINALES**

**Script de centro de computos:** Se desarrollo un script de tipo Shell en forma modular con la formación de un menú para el usuario, concebido para la automatización de tareas en el sistema operativo. El script, aborda aspectos cruciales del sistema, proporcionando información detallada sobre intentos de inicio de sesión, estadísticas del sistema y servicios activos. A continuación, se ofrece un análisis detallado de cada función implementada.

**Menu:** Dicho menú permitirá al usuario el uso correcto e intuitivo de dicha herramienta pudiendo facilitarle así la búsqueda de una tarea en concreto a realizar.



**Funcionalidades.**

**1. Función inicio de sesión fallidos:**

La función detalle fallidos se centra en proporcionar estadísticas detalladas sobre los intentos de inicio de sesión fallidos. A través del uso de sudo grep, busca patrones específicos en el archivo definido por $RUTA. La salida incluye el número total de intentos fallidos y detalles puntuales sobre los usuarios afectados. Este enfoque ofrece una visión integral de la seguridad del sistema, permitiendo la identificación temprana de posibles amenazas.

**2. Función inicio de sesión exitosos:**

La función detalle exitosos aborda los intentos de inicio de sesión exitosos utilizando también sudo grep. La salida presenta el número total de intentos exitosos y proporciona detalles específicos sobre los usuarios involucrados. Este análisis contribuye a una comprensión completa de la actividad de inicio de sesión autorizada, fundamental para evaluar la salud del sistema.

**3. Función log inicio de sesión completo:**

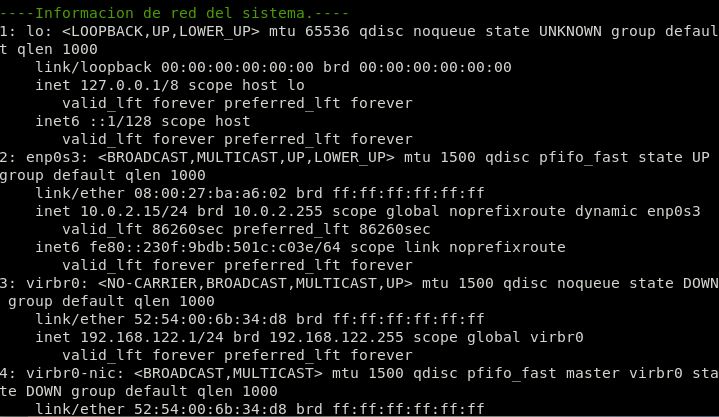
La función detalle completo integra las funciones previas, proporcionando una visión completa de los intentos de inicio de sesión, ya sean fallidos o exitosos. Este enfoque consolidado facilita la revisión de la actividad de inicio de sesión en un solo informe, simplificando la identificación de patrones y comportamientos irregulares.

**4. Función centros de computos:**

La función centro de computos se dedica a proporcionar información detallada sobre el sistema y el uso de recursos. Desde el nombre del host hasta detalles sobre la CPU y la memoria, esta función crea un informe comprensible que facilita el monitoreo del rendimiento del sistema.

**5. Función red:**

La función red se centra en la información de red del sistema, utilizando el comando ip addr. Este análisis detallado de las interfaces de red proporciona una instantánea útil para entender la configuración de red del sistema.



A continuación su explicación de cada interfaz mostrada en la imagen.

**Interfaz lo (Loopback):**

Función: La interfaz lo es esencialmente el loopback, permitiendo que el sistema se comunique consigo mismo. Su propósito principal es facilitar la comunicación interna dentro del propio dispositivo, sin pasar por la red física.

**Interfaz enp0s3:**

Función: La interfaz enp0s3 representa la conexión de red principal del sistema. Esta interfaz está activa y configurada para utilizar difusión (BROADCAST). Su función es proporcionar conectividad a la red física, permitiendo la comunicación con otros dispositivos en la misma red.

**Interfaz virbr0:**

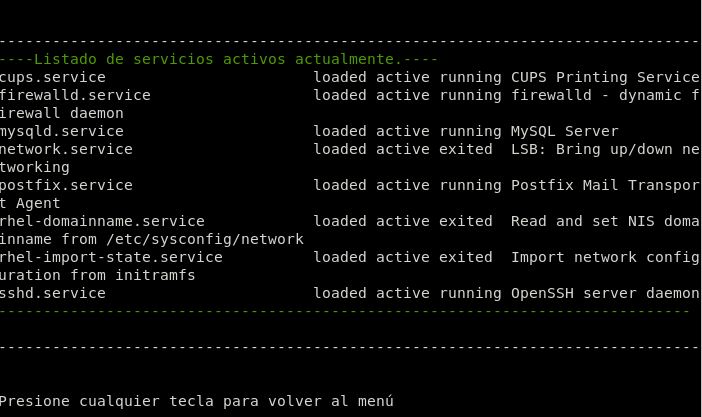
Función: La interfaz virbr0 es una interfaz virtual asociada a la virtualización. Actualmente, se encuentra desconectada (DOWN). Su propósito principal es facilitar la comunicación entre las máquinas virtuales y con el host mediante una red interna virtual.

**Interfaz virbr0-nic:**

Función: La interfaz virbr0-nic está vinculada a virbr0 y actualmente se encuentra desconectada (DOWN). Sirve como interfaz puente para la interfaz virtual virbr0, facilitando la comunicación entre las máquinas virtuales y la red física

**6. Función servicios activos:**

La función servicios activos lista los servicios activos en el sistema, focalizándose en servicios comunes. Utiliza el comando systemctl list-units y grep para presentar de manera clara los servicios relevantes para el monitoreo y gestión del sistema. A continuación se procede a la explicación y funcionamiento de dichos servicios esenciales.



**CUPS Printing Service:**

Funcionamiento: El servicio CUPS (Common Unix Printing System) facilita la gestión de impresión en sistemas Unix. Su estado "loaded active running" indica que está cargado, activo y en ejecución. Permite la impresión de documentos en entornos compartidos, gestionando colas de impresión y proporcionando una interfaz para la configuración de impresoras.

Firewalld - Dynamic Firewall Daemon:

Funcionamiento: El servicio Firewalld es un demonio de firewall dinámico. Su estado "loaded active running" indica que está cargado, activo y en ejecución. Firewalld gestiona las reglas de firewall de forma dinámica, permitiendo ajustes en tiempo real. Es esencial para la seguridad del sistema, controlando el tráfico de red según las políticas definidas.

**MySQL Server:**

Funcionamiento: El servicio mysqld corresponde al servidor MySQL. Su estado "loaded active running" señala que está cargado, activo y en ejecución. MySQL es un sistema de gestión de bases de datos relacional utilizado para almacenar y recuperar datos. El servicio asegura que la base de datos esté disponible y funcionando.

**LSB: Bring up/down networking:**

Funcionamiento: El servicio network es responsable de gestionar la red del sistema. Su estado "loaded active exited" indica que está cargado, activo y ha finalizado su ejecución. Este servicio controla la inicialización y desactivación de la red, asegurando una conexión estable y confiable.

**Postfix Mail Transport Agent:**

Funcionamiento: El servicio Postfix es un Agente de Transporte de Correo (MTA) utilizado para la gestión y entrega de correos electrónicos. Su estado "loaded active running" indica que está cargado, activo y en ejecución. Postfix facilita el envío y recepción de correos electrónicos en el sistema.

**OpenSSH server daemon:**

Funcionamiento: El servicio sshd corresponde al daemon del servidor OpenSSH. Su estado "loaded active running" indica que está cargado, activo y en ejecución. SSH (Secure Shell) permite la conexión segura a través de la red y facilita la administración remota del sistema. El servicio sshd asegura la disponibilidad del servidor SSH para conexiones seguras.

**Análisis final.**

Este script brinda un conjunto integral de funciones que abordan áreas clave del sistema operativo. La implementación de estas funciones proporciona una automatización efectiva para la recopilación de datos y el monitoreo del sistema. La presentación clara y detallada de la información facilita la interpretación de la actividad del sistema, mejorando así la capacidad de respuesta y la seguridad general.

**Script de instalación y backups:**

Este playbook de Ansible tiene como objetivo simplificar y automatizar el proceso de instalación y configuración del servidor MySQL, también así como la instalación de un firewall y puesta en marcha, además de contar con backups a la base de datos que son realizados periódicamente en conjunto a los scripts y/o archivos utilizado en el sistema en su entorno local. A continuación, se presentan detalladamente las funciones de cada tarea.

**Tareas a ejecutar.**

**Actualización de paquetes**: Esta tarea se encarga de mantener el sistema actualizado mediante el uso del módulo yum. De esta manera, garantiza que todas las dependencias y paquetes del sistema estén en su versión más reciente, mejorando la seguridad y estabilidad del entorno.

**Instalación MySQL**: En esta tarea, se emplea el módulo yum para instalar el paquete mysql-server. Esto asegura que MySQL, un sistema de gestión de bases de datos, esté presente en el servidor. La instalación de MySQL es esencial para habilitar la persistencia de datos y facilitar su manipulación.

**Inicio y habilitación de servicio MySQL**: Utilizando el módulo systemd, esta tarea inicia y habilita el servicio MySQL. Al hacerlo, aseguramos que el servidor de bases de datos esté en ejecución y configurado para arrancar automáticamente con el sistema. Esto garantiza la disponibilidad continua de MySQL.

**Instalación de firewalld**: En esta fase, se utiliza el módulo yum para instalar el paquete firewalld. Este es un componente crucial para la seguridad del sistema, ya que proporciona una interfaz para gestionar el firewall de manera eficiente. La instalación de firewalld sienta las bases para configuraciones avanzadas de reglas de seguridad.

**Inicio y habilitación firewall-d**: Utilizando el módulo systemd, esta tarea inicia el servicio firewalld y lo configura para iniciarse automáticamente con el sistema. El firewall es esencial para proteger el sistema contra posibles amenazas externas. El inicio automático garantiza que la protección esté activa desde el inicio del sistema.

**Apertura de puerto 3306 en el firewall para MySQL**: En esta tarea, se utiliza el módulo firewalld para abrir de manera permanente el puerto 3306 en el firewall. Este paso es crucial para permitir conexiones a MySQL desde otras máquinas. La configuración permanente asegura que el puerto siga abierto incluso después de reinicios del sistema, garantizando la accesibilidad continua al servicio de bases de datos.

**Ejecución de backup de documentación**: En este caso se ejecuta un script de respaldo de documentación que contiene scripts y/o archivos de relevancia, Dicho script se encuentra en la ruta "/root/Escritorio/script\_backup\_final/**Backups\_rutina.sh**". Este proceso asegura la preservación de la documentación crítica del sistema además de realizar resplados rutininarios a traves de él uso de crontab.

**Ejecución backup para la base de datos**: Se ejecuta un script de respaldo de la base de datos cuando es activado por alguna tarea. En este caso el script se encuentra en la ruta "/root/Escritorio/script\_backup\_final/**Backup\_bdd.sh**". Este respaldo garantiza la integridad y disponibilidad de los datos almacenados en MySQL.