# Usuarios

## 1. Crear Usuarios

### 1. useradd

useradd es el comando de bajo nivel para agregar usuarios. Viene por defecto en la mayoría de las distribuciones de Linux y ofrece una forma más básica y manual de crear usuarios. Al usar useradd, generalmente necesitas especificar todas las opciones manualmente, como la creación del directorio home, la selección de la shell, entre otros.

Es parte del paquete passwd.

No crea directorios home por defecto (a menos que se especifique con la opción -m).

Generalmente requiere más opciones y argumentos para configurar un usuario completamente (como la shell, directorio home, etc.).

Por ejemplo, para crear un usuario con useradd, podrías necesitar un comando como:

sudo useradd -m -s /bin/bash nuevo\_usuario

### 2. adduser

adduser, por otro lado, es un script de Perl más amigable para el usuario que simplifica el proceso de agregar usuarios. Este comando es más interactivo y automatiza muchos de los pasos que requiere useradd.

Es un script frontal que eventualmente llama a useradd en el fondo, pero automatiza procesos como la creación del directorio home y la asignación de una shell.

Proporciona un proceso más interactivo, solicitando al administrador ingresar detalles como la contraseña, nombre completo del usuario, y otros datos opcionales.

Generalmente, es más seguro para los principiantes porque guía al usuario a través del proceso y reduce el riesgo de omitir pasos importantes o cometer errores.

Un ejemplo típico de uso de adduser sería simplemente:

sudo adduser nuevo\_usuario

Este comando inicia un proceso interactivo que te guiará a través de la configuración del nuevo usuario.

Conclusión

La elección entre adduser y useradd depende en gran medida de tus necesidades específicas y tu nivel de comodidad con la línea de comandos:

useradd es mejor para scripts y automatización, donde necesitas control total sobre cómo se configuran las cuentas de usuario.

adduser es preferible para administradores que prefieren una herramienta más interactiva y menos propensa a errores para la creación de usuarios en sistemas donde está disponible.

Para crear un nuevo usuario en Linux, puedes utilizar el comando useradd. A continuación, te muestro cómo usarlo:

sudo useradd [opciones] nombre\_de\_usuario

[opciones]: Aquí puedes añadir opciones como -m para crear el directorio home del usuario, -s para especificar la shell que usará, entre otras.

nombre\_de\_usuario: Es el nombre del usuario que deseas crear.

Por ejemplo, para crear un usuario llamado juan con un directorio home y bash como shell, usarías:

sudo useradd -m -s /bin/bash juan

Después de crear un usuario, generalmente querrás establecer o cambiar su contraseña. Para eso, utiliza el comando passwd:

sudo passwd juan

Se te pedirá que ingreses y confirmes la contraseña.

## 2. Gestionar Usuarios

Cambiar el Shell de un Usuario

Si deseas cambiar la shell de un usuario, puedes usar el comando usermod. Por ejemplo, para cambiar la shell de juan a zsh, podrías hacer:

sudo usermod -s /bin/zsh juan

Añadir un Usuario a un Grupo

Para añadir un usuario a un grupo adicional, puedes usar también usermod:

sudo usermod -aG nombre\_del\_grupo juan

## 3. Eliminar Usuarios

Para eliminar un usuario, el comando a utilizar es userdel. Aquí te muestro cómo:

sudo userdel nombre\_de\_usuario

Si también quieres eliminar el directorio home del usuario y su correo, puedes añadir la opción -r:

sudo userdel -r juan

Este comando eliminará al usuario juan junto con su directorio home y su correo.

Consideraciones Finales

Siempre asegúrate de tener privilegios de superusuario cuando realices estos comandos, de ahí el uso de sudo.

Cada cambio que hagas en la gestión de usuarios es importante y puede afectar la seguridad y la operatividad del sistema. Asegúrate de saber lo que estás haciendo antes de ejecutar estos comandos.

# Grupos

## 1. Crear Grupos

### con groupadd

groupadd es el comando para crear un nuevo grupo en el sistema. Aquí te muestro cómo usarlo:

sudo groupadd nombre\_del\_grupo

nombre\_del\_grupo: Es el nombre del grupo que deseas crear.

Este comando crea un nuevo grupo sin usuarios. Para agregar usuarios a este grupo, necesitas utilizar el comando usermod o gpasswd.

### Con addgroup

addgroup, por otro lado, es un script más amigable e interactivo que puede facilitar la creación de grupos, especialmente para los usuarios que no están familiarizados con todas las opciones disponibles. Este comando es más común en distribuciones basadas en Debian y es un frontend para groupadd.

Interactivo: En algunos casos, addgroup puede ser interactivo, guiando al administrador a través del proceso de creación del grupo y solicitando información adicional si es necesario.

Automatización: Puede manejar más configuraciones automáticamente, como añadir usuarios al grupo durante el proceso de creación.

Ejemplo de uso:

sudo addgroup nuevo\_grupo

## 2. Modificar Grupos con groupmod

groupmod permite modificar las propiedades de un grupo existente. Puedes cambiar el nombre del grupo o su ID (GID) usando este comando. Aquí hay algunos ejemplos de cómo usar groupmod:

Cambiar el nombre de un grupo:

sudo groupmod -n nuevo\_nombre nombre\_del\_grupo

Cambiar el GID de un grupo:

sudo groupmod -g nuevo\_gid nombre\_del\_grupo

[nuevo\_nombre]: Es el nuevo nombre que quieres dar al grupo.

[nuevo\_gid]: Es el nuevo ID numérico que quieres asignar al grupo.

## 3. Eliminar Grupos con groupdel

Para eliminar un grupo del sistema, puedes usar groupdel. Es importante asegurarse de que ningún usuario esté utilizando el grupo como su grupo principal antes de eliminarlo. Aquí está el comando básico:

sudo groupdel nombre\_del\_grupo

## 4. Añadir o Eliminar Usuarios de Grupos

Además de los comandos específicos para gestionar grupos, es común necesitar añadir o eliminar usuarios de grupos:

### Añadir un usuario a un grupo:

**sudo usermod -aG nombre\_del\_grupo nombre\_de\_usuario**

Este comando añade el usuario al grupo sin eliminarlo de otros grupos a los que ya pertenece.

**sudo addgroup juan desarrolladores**

Este comando no solo verificará si el grupo existe, sino que también añadirá al usuario juan al grupo desarrolladores si el usuario existe.

### Eliminar un usuario de un grupo:

Puedes utilizar gpasswd para este propósito:

**sudo gpasswd -d nombre\_de\_usuario nombre\_del\_grupo**

Este comando elimina el usuario del grupo especificado.

# Casos específicos

## **Crear un Grupo y Añadir un Usuario en un Solo Paso**

En algunas distribuciones, addgroup permite crear un nuevo grupo y añadir un usuario a este grupo en un solo paso. Esto se puede hacer durante la creación del usuario con el comando adduser especificando el grupo directamente:

**sudo adduser nuevo\_usuario nuevo\_grupo**

## Listar Miembros de un Grupo

**getent group nombre\_del\_grupo**

## Comando chage (cambiar contraseña)

El comando chage en Linux es una herramienta poderosa para gestionar la política de envejecimiento de las contraseñas de los usuarios. Su nombre proviene de "change age" (cambiar edad), y su uso principal es modificar y ver la información de caducidad de las contraseñas y cuentas de los usuarios. Este comando es esencial para mantener una política de seguridad adecuada en entornos que requieren el cambio regular de contraseñas para mejorar la seguridad.

Funciones Principales de chage

Establecer la Fecha de Expiración de la Contraseña: Puedes definir cuándo una contraseña debe expirar, forzando al usuario a cambiarla después de cierto tiempo.

Modificar las Propiedades de Envejecimiento de la Contraseña: Ajusta las políticas relacionadas con cuánto tiempo una contraseña puede y debe durar antes de necesitar ser cambiada.

Ver la Información de Envejecimiento de la Contraseña: Permite a los administradores ver la configuración actual de envejecimiento de las contraseñas para cualquier usuario.

Ejemplos de Uso del Comando chage

Aquí te muestro algunos ejemplos comunes de cómo se puede usar chage:

### Ver la Información de Envejecimiento de una Contraseña

Para ver la información actual del envejecimiento de la contraseña de un usuario, puedes utilizar:

**sudo chage -l nombre\_de\_usuario**

Este comando mostrará información como la fecha del último cambio de contraseña, cuándo expirará la contraseña, y cuándo la cuenta será desactivada.

### Establecer la Caducidad de la Contraseña

Para establecer o cambiar cuándo debe expirar la contraseña de un usuario:

sudo chage -M dias nombre\_de\_usuario

Donde dias es el número de días después del último cambio de contraseña en el que la contraseña del usuario expirará.

### Establecer la Cuenta para que Expira

Puedes especificar una fecha en la que la cuenta del usuario será desactivada completamente:

sudo chage -E fecha nombre\_de\_usuario

La fecha debe estar en el formato AAAA-MM-DD.

### Establecer el Mínimo de Días Antes del Cambio de Contraseña

Para establecer el número mínimo de días que deben pasar antes de que un usuario pueda cambiar su contraseña:

sudo chage -m dias nombre\_de\_usuario

### Establecer el Máximo de Días Antes del Cambio de Contraseña

Para definir el número máximo de días durante los cuales la contraseña del usuario será válida:

sudo chage -M dias nombre\_de\_usuario

### Establecer un Periodo de Advertencia Antes de la Caducidad de la Contraseña

Para especificar cuántos días antes de la caducidad de la contraseña se comenzará a advertir al usuario que debe cambiarla:

sudo chage -W dias nombre\_de\_usuario

Importancia de chage

El comando chage es crucial en entornos que manejan datos sensibles o donde se requiere una alta seguridad, ya que permite a los administradores asegurar que las contraseñas sean cambiadas regularmente. Esta es una práctica recomendada para evitar accesos no autorizados debido a contraseñas comprometidas o débiles que no se cambian con frecuencia.

# Permisos

Para modificar los permisos de archivos y directorios en Linux, utilizamos principalmente dos comandos: chmod (change mode) y chown (change owner). Estos comandos permiten a los administradores y usuarios controlar quién puede acceder y modificar archivos y directorios en el sistema. Aquí te muestro cómo utilizar cada uno de estos comandos para cambiar permisos de usuario y grupo.

1. chmod - Cambiar Permisos de Archivos y Directorios

El comando chmod se utiliza para cambiar los permisos de archivo para los propietarios (usuario), grupo y otros (otros usuarios). Los permisos pueden ser ajustados usando un modo numérico o simbólico.

Modo Numérico

En el modo numérico, los permisos se establecen usando números que representan los diferentes niveles de permisos:

4 - Lectura (r)

2 - Escritura (w)

1 - Ejecución (x)

Los permisos se especifican como tres dígitos, representando al propietario, al grupo y a otros, respectivamente. Por ejemplo:

sudo chmod 755 archivo

Este comando otorga lectura, escritura y ejecución al propietario (7), y lectura y ejecución a grupo y otros (5 y 5).

Modo Simbólico

También puedes usar letras y símbolos para definir los permisos:

u = usuario (propietario)

g = grupo

o = otros

a = todos (u, g, o)

Ejemplo para agregar permiso de ejecución al grupo:

sudo chmod g+x archivo

Ejemplo para remover permisos de escritura para otros:

sudo chmod o-w archivo

2. chown - Cambiar Propietario y Grupo de Archivos y Directorios

El comando chown permite cambiar el propietario y el grupo de un archivo o directorio. Esto es útil cuando necesitas transferir la propiedad de archivos o directorios a otro usuario o grupo.

Cambiar Propietario

Para cambiar el propietario de un archivo:

sudo chown nuevo\_propietario archivo

Cambiar Propietario y Grupo

Para cambiar tanto el propietario como el grupo al mismo tiempo:

sudo chown nuevo\_propietario:nuevo\_grupo archivo

Cambiar Grupo

Si solo necesitas cambiar el grupo de un archivo:

sudo chown :nuevo\_grupo archivo

Aplicar Recursivamente

Si necesitas cambiar los permisos o propietarios de manera recursiva dentro de directorios, puedes añadir la opción -R:

sudo chown -R nuevo\_propietario:nuevo\_grupo directorio

sudo chmod -R 755 directorio

Estos comandos te permitirán administrar efectivamente los permisos y la propiedad de archivos y directorios en tu sistema Linux, asegurando que solo los usuarios y grupos adecuados tengan acceso a los recursos necesarios.