

Bootcamp Devops Engineer

Encuentro #7

Antes de empezar... quiz!

Que vamos a ver hoy?

- Q&A Desafío
- Gestión de Usuarios y Grupos
- Política de contraseñas y cuentas



Q&A

Gestión de Usuarios y Grupos



Iniciar nuestra instancia Linux

- Crear la instancia - `multipass launch -n <nombre>`
- Tomar la shell de nuestra instancia - `multipass shell <nombre>`

Clonar repositorio de la clase

- `git clone https://github.com/JCaimo/Clase7.git`

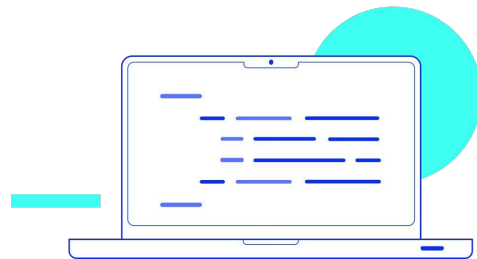
Introducción a la Gestión de Usuarios

Localización de Cuentas de Usuario:

- Archivo `/etc/passwd` para cuentas de usuario.
- Archivo `/etc/shadow` para contraseñas cifradas.

Creación de Cuentas de Usuario:

- Comando `useradd`
- Uso del directorio plantilla `/etc/skel`

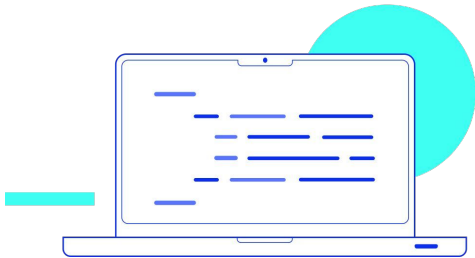


Archivo /etc/passwd

Este archivo es crucial para la administración de usuarios en sistemas Linux. Contiene información sobre cada usuario del sistema en líneas separadas, donde cada línea representa una cuenta de usuario.

Formato:

- `nombre_usuario:contraseña:x:UID:GID:información_usuario:directorio_inicio:shell`

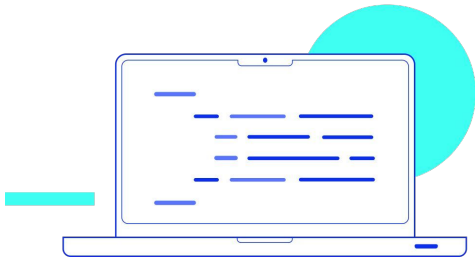


Archivo /etc/shadow

Este archivo almacena las contraseñas cifradas y otra información de seguridad relacionada con las cuentas de usuario.

Formato:

- nombre_usuario:contraseña_cifrada:
último_cambio:min_días:max_días:aviso:i
nactivo:expiración:reservado



Comando: useradd

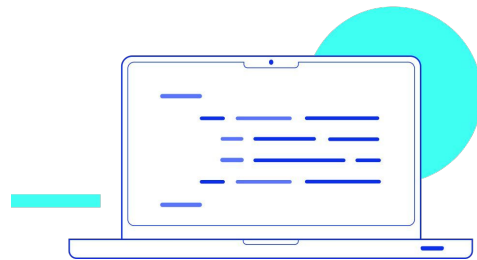
El comando `useradd` es utilizado para crear nuevas cuentas de usuario en el sistema.

Sintaxis básica:

- `useradd [opciones] nombreDelUsuario`
- Ejemplo: `useradd usuario1`

Opciones:

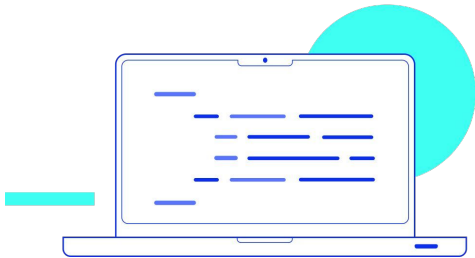
- Grupo Primario: `useradd -g sistemas usuario1`
- Fecha de Expiración de la Cuenta: `useradd -e 2024-12-31 usuario3`
- Múltiples Grupos Secundarios: `useradd -G docker, ansible usuario4`



Comando: useradd

Opciones:

- Usuario del Sistema: `useradd -r backup`
- Usuario con un Identificador de Usuario Específico: `useradd -u 1050 usuario5`
- Usuario con un Shell Específico: `useradd -s /bin/zsh usuario6`
- Usuario con Home y Copiar Contenido de /etc/skel: `useradd -m usuario7`



Directorio: /etc/skel

El directorio `/etc/skel` (skeleton) es un directorio plantilla que contiene archivos y directorios predeterminados que se copian al directorio home de un nuevo usuario cuando se crea su cuenta.

Propósito de `/etc/skel`:

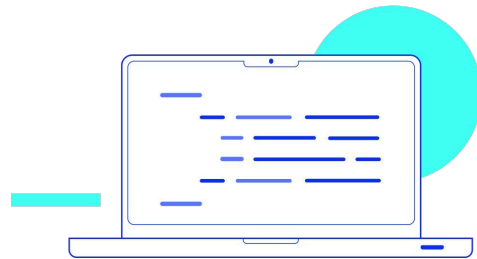
- Asegura que cada nuevo usuario tenga un conjunto básico y consistente de archivos y configuraciones desde el inicio.
- Facilita la configuración de un entorno de trabajo uniforme para todos los usuarios.



Directorio: /etc/skel

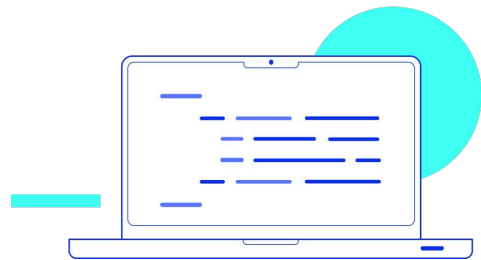
Contenido Típico de /etc/skel:

- **.bashrc:** Script de configuración del shell Bash que se ejecuta para sesiones interactivas.
- **.profile:** Script de configuración del shell que se ejecuta para sesiones de login.
- **.bash_logout:** Script que se ejecuta cuando una sesión de Bash termina.
- **Directorios Predefinidos:** Pueden incluir directorios como Documents, Downloads, Desktop, etc.



Comandos: id, usermod, userdel y passwd

- Ver información del usuario actual: `id`
- Ver información de un usuario específico: `id juan`
- Cambiar el Directorio Home de un Usuario: `usermod -d /nuevo/home/juan -m juan`
- Cambiar el UID de un Usuario: `usermod -u 2000 juan`
- Eliminar un Usuario: `userdel pedro`
- Eliminar un Usuario y su Directorio Home: `userdel -r pedro`
- Cambiar la Contraseña de un Usuario: `passwd juan`
- Forzar a un Usuario a Cambiar su Contraseña en el Próximo Inicio de Sesión: `passwd -e juan`



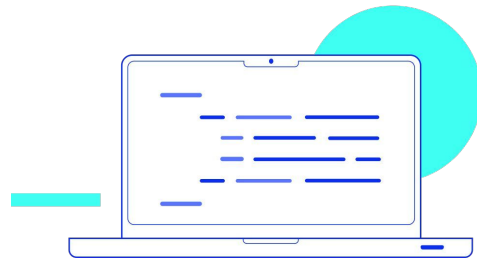
Introducción a la Gestión de Grupos

Localización de Grupos:

- Archivo `/etc/group` para cuentas de usuario.

Administración de grupos:

- Comando `groupadd`
- Comando `groupmod`

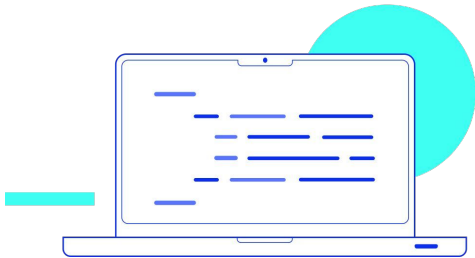


Archivo /etc/group

El archivo `/etc/group` es donde se almacenan todos los grupos del sistema. Cada línea en este archivo representa un grupo y contiene información sobre el nombre del grupo, su contraseña (si tiene), el GID y los miembros del grupo.

Formato:

- `nombre_del_grupo:contraseña:GID:lista_de_usuarios`



Comando: groupadd

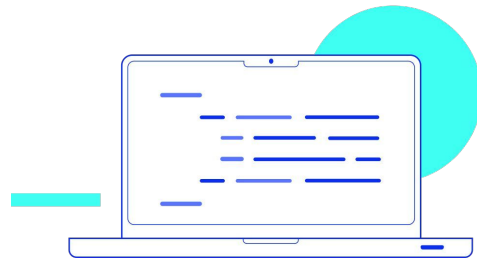
El comando chage se utiliza para listar o cambiar el tiempo de expiración de una contraseña de usuario.

Sintaxis básica:

- chage [opciones] nombreDelUsuario
- Ejemplo: `groupadd grupo1`

Opciones:

- Definir GID específico: `groupadd -g 1001 desarrolladores`
- Forzar la creación: `groupadd -f ventas`
- Crear grupo del sistema: `groupadd -r servicios`



Comando: groupmod

El comando groupmod se utiliza para modificar las propiedades de los grupos existentes en el sistema Linux.

Sintaxis básica:

- `groupmod [opciones] nombreDelGrupo`

Opciones:

- Cambiar GID: `groupmod -g 1050 desarrolladores`
- Cambiar nombre: `groupmod -n programadores desarrolladores`



Política de Contraseñas y Cuentas

Políticas de Contraseñas y Cuentas

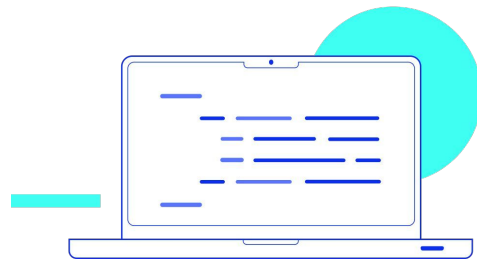
Las políticas de contraseñas y cuentas son fundamentales para la seguridad en sistemas Linux, permitiendo un control efectivo de los usuarios.

Terminología:

- **Política de Contraseña:** Reglas que dictan la gestión y cambios de contraseñas de los usuarios.

Importancia:

- Protege la integridad y confidencialidad de los datos.
- Previene accesos no autorizados y potenciales brechas de seguridad.



Comando: chage

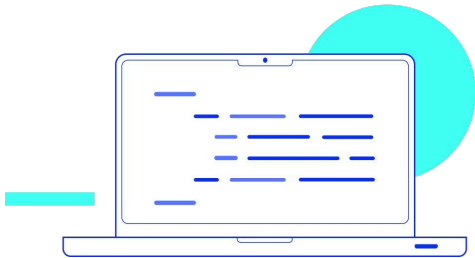
El comando `chage` se utiliza para listar o cambiar el tiempo de expiración de una contraseña de usuario.

Sintaxis básica:

- `chage [opciones] nombreDelUsuario`

Opciones:

- Modificar días desde último cambio de contraseña: `chage -d días`
- Modificar fecha de expiración de cuenta: `chage -E fecha`
- Modificar días antes de bloquear cuenta expirada `chage -l días`



Comando: chage

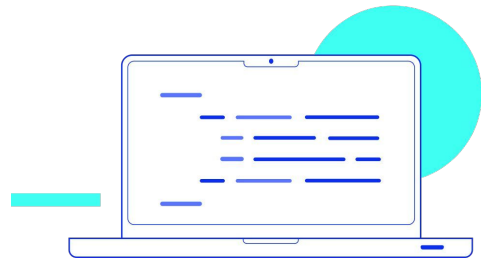
Opciones:

- Modificar días máximo de validez de contraseña: `chage -M días`
- Modificar días mínimo entre cambios de contraseña: `chage -m días`
- Modificar días de aviso antes de expiración: `chage -W días`
- Mostrar información del usuario: `chage -l usuario`
- Como automatizamos esto?



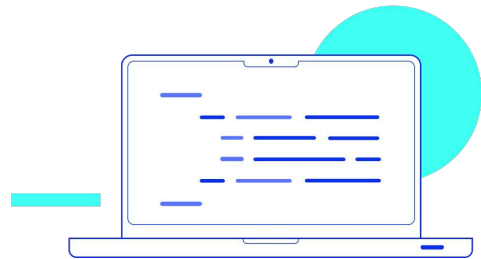
Ejercicios!

- Crear un usuario con un grupo primario específico
- Crear un usuario con directorio home personalizado
- Agregar un usuario a múltiples grupos secundarios
- Modificar el directorio home de un usuario
- Eliminar un usuario y su directorio home
- Forzar a un usuario a cambiar su contraseña en el próximo inicio de sesión
- Configurar la expiración de la contraseña de un usuario para que expire en 30 días



Ejercicios!

- Crea un usuario llamado `usuario_avanzado` con las siguientes características:
 - Grupo primario desarrollo
 - Directorio home `/home/usuario_avanzado`
 - Fecha de expiración de cuenta `2025-01-01`
 - Shell `/bin/zsh`
 - UID `1050`
 - Miembro de los grupos secundarios `docker` y `ansible`
- Migración de un usuario a un nuevo UID y grupo llamado operaciones



Ejercicios!

- Configura el sistema para que todas las contraseñas deben:
 - Tener al menos 12 caracteres
 - Incluir al menos una letra mayúscula, una minúscula, un número y un carácter especial
 - Ser cambiadas cada 90 días

