

Gestión avanzada de permisos de usuarios y grupos en contenedores Docker con Ubuntu

Karen Andrade Gallardo
Universidad de Cuenca
Ingeniería en Telecomunicaciones
karen.andradeg@ucuenca.edu.ec

Resumen—Este informe tiene como objetivo principal implementar un sistema avanzado de gestión de permisos de usuarios y grupos dentro de un contenedor Ubuntu. Se crearon usuarios y grupos con UID/GID específicos, se configuraron directorios home con archivos personales y tareas, se establecieron permisos restrictivos y se verificó el correcto aislamiento entre usuarios. Finalmente, se generó un resumen completo de usuarios, grupos y permisos y estructura de directorios, demostrando el control granular que ofrece Linux dentro de entornos contenedorizados.

Index Terms—Docker, Contenedores, Linux, Permisos, Usuarios, Grupos, Ubuntu, Seguridad

I. INTRODUCCIÓN

El uso de contenedores Docker permite ejecutar aplicaciones aisladas de forma ligera y portable. Sin embargo, en muchos escenarios reales es necesario replicar dentro del contenedor una estructura completa de usuarios y grupos similar a la de un sistema operativo tradicional, con el fin de aplicar políticas de seguridad, separación de privilegios y cumplimiento de requisitos organizacionales.

El Reto 5.4 consistió en crear y configurar dentro de un contenedor Ubuntu los siguientes elementos:

- Usuarios: Supervisor (UID 1001), Mantenimiento (UID 1002), Profesor (UID 1003), Alumno (UID 1004).
- Grupos secundarios: Tecnología (GID 1020), Colegio (GID 1030).
- Estructura de directorios /home con permisos estrictos.
- Archivos de tareas (Tarea-A.txt, Tarea-B.txt, Tarea-C.txt) y archivo CONTRATO.txt exclusivos de ciertos usuarios.
- Verificación de aislamiento entre usuarios.

II. MARCO TEÓRICO

II-A. Conceptos fundamentales

- **Contenerización:** tecnología que permite empaquetar una aplicación con todas sus dependencias en una unidad estandarizada llamada contenedor.
- **Docker:** plataforma líder para creación, distribución y ejecución de contenedores.
- **Imágenes Docker:** plantillas de solo lectura que contienen el sistema de archivos y metadatos necesarios para crear contenedores.
- **Contenedores:** instancias ejecutables de una imagen.
- **UID/GID:** identificadores numéricos de usuario y grupo, en sistemas Unix/Linux.

- **/etc/passwd y /etc/group:** archivos que almacenan la información de usuarios y grupos.

II-B. Tabla de comandos utilizados

Cuadro I
PRINCIPALES COMANDOS EMPLEADOS

Comando	Descripción
useradd, usermod	Crear y modificar usuarios
groupadd, groupmod	Crear y modificar grupos
chown, chgrp	Cambiar propietario y grupo de archivos
chmod	Cambiar permisos de archivos y directorios
mkdir -p	Crear estructura de directorios
touch	Crear archivos vacíos
echo	Escribir contenido en archivos
su - usuario	Cambiar a otro usuario
tree	Mostrar estructura de árbol
cat /etc/passwd — grep	Filtrar usuarios

III. DESARROLLO

III-A. Creación y configuración del contenedor

Se ejecutó el contenedor interactivo con la imagen pbarbecho/ubuntu_permisos:latest:

```
docker run -it --name permisos-container \
pbarbecho/ubuntu_permisos:latest /bin/bash
```

Listing 1. Inicio del contenedor

Dentro del contenedor se instaló el comando tree:

```
apt update && apt install -y tree
```

III-B. Creación de usuarios y grupos

```
# Grupos
groupadd -g 1020 Tecnologia
groupadd -g 1030 Colegio

# Usuarios con UID específicos y grupos secundarios
useradd -m -u 1001 -G Tecnologia,Mantenimiento Supervisor
useradd -m -u 1002 -G Tecnologia,Mantenimiento Mantenimiento
useradd -m -u 1003 -G Colegio,Profesor Profesor
useradd -m -u 1004 -G Colegio,Alumno Alumno
```

```

11 # Contrase as (todas 1111)
12 echo "Supervisor:1111" | chpasswd
13 echo "Mantenimiento:1111" | chpasswd
14 echo "Profesor:1111" | chpasswd
15 echo "Alumno:1111" | chpasswd

```

III-C. Estructura de directorios y archivos

Se crearon los archivos de tareas y CONTRATO.txt con propietarios y permisos específicos:

```

1 # Archivos en /home/Alumno
2 install -o Alumno -g Colegio -m 644 /dev/null
  /home/Alumno/Tarea-A.txt
3 install -o Alumno -g Colegio -m 644 /dev/null
  /home/Alumno/Tarea-B.txt
4 install -o Alumno -g Colegio -m 644 /dev/null
  /home/Alumno/Tarea-C.txt
5
6 # Archivos en /home/Profesor
7 install -o Profesor -g Profesor -m 644 /dev/
  null /home/Profesor/Alumno-A.txt
8 install -o Profesor -g Profesor -m 644 /dev/
  null /home/Profesor/Alumno-B.txt
9 install -o Profesor -g Profesor -m 644 /dev/
  null /home/Profesor/Alumno-C.txt
10
11 # CONTRATO.txt exclusivo del Supervisor
12 install -o Supervisor -g Supervisor -m 600 /
  home/Supervisor/CONTRATO.txt
13 echo "Confidencial" > /home/Supervisor/
  CONTRATO.txt

```

III-D. Archivos de control en /home/Mantenimiento

```

1 echo -e "Supervisor:_1001\nMantenimiento:_
  1002\nProfesor:_1003\nAlumno:_1004" > /
  home/Mantenimiento/usuarios.txt
2
3 echo -e "Tecnologia:_1020\nColegio:_1030" > /
  home/Mantenimiento/grupos.txt
4
5 chown Mantenimiento:Mantenimiento /home/
  Mantenimiento/*.txt
6 chmod 600 /home/Mantenimiento/*.txt

```

III-E. Restricción de acceso entre usuarios

Se comprobó que el usuario Alumno no podía acceder a los directorios de otros usuarios:

```

1 su - Alumno -c "tree_/home"
2 # Resultado: solo ve su propio directorio y
  los archivos Tarea-*.txt

```

III-F. Resumen final generado

Se creó un script de resumen que muestra:

- Usuarios y sus directorios home/shell
- Grupos creados
- Listado completo de /home con permisos
- Contenido de usuarios.txt y grupos.txt

III-G. Capturas de Pantalla

Figura 1. Captura No.1

Figura 2. Captura No.2

Figura 3. Captura No.3

Figura 4. Captura No.4

```

root@88c2238902d2:/# echo "=== 1001:201800L_L3R1TAD00 (Alumno) ==="
ARbol_L3R1TAD00 (Alumno) ===
root@88c2238902d2:/# su - Alumno -c "tree /home 2>/dev/null || echo 'Acceso restringido a otros directorios'"
/home
├── Alumno
│   ├── Tarea-A.txt
│   ├── Tarea-B.txt
│   └── Tarea-C.txt
├── Mantenimiento [error opening dir]
├── Profesor [error opening dir]
├── Supervisor [error opening dir]
└── ubuntu [error opening dir]

6 directories, 3 files
Acceso restringido a otros directorios
root@88c2238902d2:/# echo "=== usuarios.txt ==="
=== usuarios.txt ===
root@88c2238902d2:/# cat /home/Mantenimiento/usuarios.txt
Supervisor: 1001
Mantenimiento: 1002
Profesor: 1003
Alumno: 1004
root@88c2238902d2:/# echo "=== grupos.txt ==="
=== grupos.txt ===
root@88c2238902d2:/# cat /home/Mantenimiento/grupos.txt
Tecnologia: 1020
Colegio: 1030
root@88c2238902d2:/# echo "=== RESUMEN COMPLETO DEL RETO 5.4 ===" && \
> echo "USUARIOS:" && cat /etc/passwd | grep -E "Supervisor|Mantenimiento|Profesor|Alumno" && \
> exit
=== RESUMEN COMPLETO DEL RETO 5.4 ===
USUARIOS:
Supervisor:x:1001:1001::/home/Supervisor:/bin/sh
Mantenimiento:x:1002:1002::/home/Mantenimiento:/bin/sh
Profesor:x:1003:1003::/home/Profesor:/bin/sh
Alumno:x:1004:1004::/home/Alumno:/bin/sh
exit
PS C:\Users\Usuario> echo "=== RESUMEN COMPLETO DEL RETO 5.4 ===" && \
&& line1 chcp:40
&& echo "=== RESUMEN COMPLETO DEL RETO 5.4 ===" && \
&& echo "=== RESUMEN COMPLETO DEL RETO 5.4 ===" && \
The token '&&' is not a valid statement separator in this version.

```

Figura 5. Captura No.5

```

PS C:\Users\Usuario> docker start -it permisos-container
unknown shorthand flag: 't' in -t

Usage: docker start [OPTIONS] CONTAINER [CONTAINER...]

Run 'docker start --help' for more information
PS C:\Users\Usuario> docker start permisos-container
permisos-container
PS C:\Users\Usuario> docker start -it permisos-container
unknown shorthand flag: 't' in -t

Usage: docker start [OPTIONS] CONTAINER [CONTAINER...]

Run 'docker start --help' for more information
PS C:\Users\Usuario> docker start permisos-container
permisos-container
PS C:\Users\Usuario> docker exec -it permisos-container /bin/bash
root@88c2238902d2:/# echo "=== RESUMEN COMPLETO DEL RETO 5.4 ===" && \
> echo "USUARIOS:" && cat /etc/passwd | grep -E "Supervisor|Mantenimiento|Profesor|Alumno" && \
> echo "GRUPOS:" && getent group Tecnologia Colegio && \
> echo "usuarios.txt:" && cat /home/Mantenimiento/usuarios.txt && \
> echo "grupos.txt:" && cat /home/Mantenimiento/grupos.txt && \
> echo "/home/Alumno:" && ls -l /home/Alumno && \
> echo "/home/Profesor:" && ls -l /home/Profesor && \
> echo "/home/Supervisor:" && ls -l /home/Supervisor && \
> echo "PERMISOS TAREA:" && ls -l /home/Alumno/Tarea*.txt && \
> echo "PERMISOS CONTRATO:" && ls -l /home/Supervisor/CONTRATO.txt && \
> echo "TREE (Supervisor):" && su - Supervisor -c "sudo tree /home" && \
> echo "TREE (Alumno):" && su - Alumno -c "tree /home 2>/dev/null || echo 'Acceso denegado'"
=== RESUMEN COMPLETO DEL RETO 5.4 ===
USUARIOS:
Supervisor:x:1001:1001::/home/Supervisor:/bin/sh
Mantenimiento:x:1002:1002::/home/Mantenimiento:/bin/sh
Profesor:x:1003:1003::/home/Profesor:/bin/sh
Alumno:x:1004:1004::/home/Alumno:/bin/sh
GRUPOS:
Tecnologia:x:1020:Supervisor,Mantenimiento
Colegio:x:1030:Profesor,Alumno
usuarios.txt:
Supervisor: 1001
Mantenimiento: 1002

```

Figura 6. Captura No.6

```

Colegio:x:1030:Profesor,Alumno
usuarios.txt:
Supervisor: 1001
Mantenimiento: 1002
Profesor: 1003
Alumno: 1004
grupos.txt:
Tecnologia: 1020
Colegio: 1030
/home/Alumno:
total 12
-rw-r--r-- 1 Alumno Colegio 57 Nov 14 15:28 Tarea-A.txt
-rw-r--r-- 1 Alumno Colegio 57 Nov 14 15:28 Tarea-B.txt
-rw-r--r-- 1 Alumno Colegio 57 Nov 14 15:28 Tarea-C.txt
/home/Profesor:
total 12
-rw-r--r-- 1 Profesor Profesor 132 Nov 14 15:28 Alumno-A.txt
-rw-r--r-- 1 Profesor Profesor 57 Nov 14 15:28 Alumno-B.txt
-rw-r--r-- 1 Profesor Profesor 57 Nov 14 15:28 Alumno-C.txt
/home/Supervisor:
total 4
-rw-r--r-- 1 Supervisor Supervisor 57 Nov 14 15:33 CONTRATO.txt
PERMISOS TAREA:
-rw-r--r-- 1 Alumno Colegio 57 Nov 14 15:28 /home/Alumno/Tarea-A.txt
-rw-r--r-- 1 Alumno Colegio 57 Nov 14 15:28 /home/Alumno/Tarea-B.txt
-rw-r--r-- 1 Alumno Colegio 57 Nov 14 15:28 /home/Alumno/Tarea-C.txt
PERMISOS CONTRATO:
-rw-r--r-- 1 Supervisor Supervisor 57 Nov 14 15:33 /home/Supervisor/CONTRATO.txt
TREE (Supervisor):
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
sudo: a password is required
root@88c2238902d2:/# echo "=== 1001:201800L_COMPLETO (como root) ==="
=== ARbol_COMPLETO (como root) ===
root@88c2238902d2:/# tree /home
/home
├── Alumno
│   ├── Tarea-A.txt
│   ├── Tarea-B.txt
│   └── Tarea-C.txt
├── Mantenimiento
│   ├── grupos.txt
│   └── usuarios.txt

```

Figura 7. Captura No.7

```

=== ARbol_COMPLETO (como root) ===
root@88c2238902d2:/# tree /home
/home
├── Alumno
│   ├── Tarea-A.txt
│   ├── Tarea-B.txt
│   └── Tarea-C.txt
├── Mantenimiento
│   ├── grupos.txt
│   └── usuarios.txt
├── Profesor
│   ├── Alumno-A.txt
│   ├── Alumno-B.txt
│   └── Alumno-C.txt
├── Supervisor
│   ├── CONTRATO.txt
└── ubuntu
    ├── Alumno-A.txt
    ├── Alumno-B.txt
    ├── Alumno-C.txt
    ├── CONTRATO.txt
    ├── Tarea-A.txt
    ├── Tarea-B.txt
    └── Tarea-C.txt

6 directories, 16 files
root@88c2238902d2:/# getent group Tecnologia
Tecnologia:x:1020:Supervisor,Mantenimiento
root@88c2238902d2:/# getent group Colegio
Colegio:x:1030:Profesor,Alumno
root@88c2238902d2:/# ls -l /home/Mantenimiento
total 8
-rw-r----- 1 Mantenimiento Mantenimiento 31 Nov 14 15:32 grupos.txt
-rw-r----- 1 Mantenimiento Mantenimiento 65 Nov 14 15:32 usuarios.txt
root@88c2238902d2:/# echo "=== usuarios.txt ==="
=== usuarios.txt ===
root@88c2238902d2:/# cat /home/Mantenimiento/usuarios.txt
Supervisor: 1001
Mantenimiento: 1002
Profesor: 1003
Alumno: 1004
root@88c2238902d2:/# echo "=== grupos.txt ==="

```

Figura 8. Captura No.8

```

├── Tarea-B.txt
├── Tarea-C.txt
├── Mantenimiento
│   ├── grupos.txt
│   ├── usuarios.txt
├── Profesor
│   ├── Alumno-A.txt
│   ├── Alumno-B.txt
│   └── Alumno-C.txt
├── Supervisor
│   ├── CONTRATO.txt
└── ubuntu
    ├── Alumno-A.txt
    ├── Alumno-B.txt
    ├── Alumno-C.txt
    ├── CONTRATO.txt
    ├── Tarea-A.txt
    ├── Tarea-B.txt
    └── Tarea-C.txt

6 directories, 16 files
root@88c2238902d2:/# getent group Tecnologia
Tecnologia:x:1020:Supervisor,Mantenimiento
root@88c2238902d2:/# getent group Colegio
Colegio:x:1030:Profesor,Alumno
root@88c2238902d2:/# ls -l /home/Mantenimiento
total 8
-rw-r----- 1 Mantenimiento Mantenimiento 31 Nov 14 15:32 grupos.txt
-rw-r----- 1 Mantenimiento Mantenimiento 65 Nov 14 15:32 usuarios.txt
root@88c2238902d2:/# echo "=== usuarios.txt ==="
=== usuarios.txt ===
root@88c2238902d2:/# cat /home/Mantenimiento/usuarios.txt
Supervisor: 1001
Mantenimiento: 1002
Profesor: 1003
Alumno: 1004
root@88c2238902d2:/# echo "=== grupos.txt ==="
=== grupos.txt ===
root@88c2238902d2:/# cat /home/Mantenimiento/grupos.txt
Tecnologia: 1020
Colegio: 1030
root@88c2238902d2:/#

```

Figura 9. Captura No.9

IV. CONCLUSIONES

- Docker permite replicar la gestión de usuarios y grupos de un sistema Linux tradicional, incluyendo UID/GID personalizados y grupos secundarios.

- La combinación de `chown`, `chgrp` y `chmod` dentro del contenedor ofrece un control total sobre los permisos, logrando un aislamiento efectivo entre usuarios no privilegiados.
- El uso de archivos centrales (`usuarios.txt`, `grupos.txt`) administrados por el usuario Mantenimiento facilita auditorías y mantenimiento.
- Se verificó que un usuario normal (Alumno) no puede listar ni acceder a los directorios home de otros usuarios cuando los permisos son 700, cumpliendo con el principio de menor privilegio.

REFERENCIAS

- [1] Docker Inc., “Docker Documentation”, *Docker.com*, 2025. [Online]. Disponible: <https://docs.docker.com/>
- [2] Canonical Ltd., “Ubuntu Server Documentation”, *Ubuntu.com*, 2025. [Online]. Disponible: <https://ubuntu.com/server/docs>
- [3] M. Kerrisk, *The Linux Programming Interface*, No Starch Press, 2010.
- [4] J. Turnbull, *The Docker Book: Containerization is the new virtualization*, 2020.